



**T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ALGORİTMA EĞİTİMİ İÇİN BİR MOBİL UYGULAMA
TASARIMI VE GERÇEKLEŞTİRİLMESİ**

FATİH DİNÇ

**YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**DANIŞMAN
DR. ÖĞR. ÜYESİ FATİH KAYAALP**

DÜZCE, 2018

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ALGORİTMA EĞİTİMİ İÇİN BİR MOBİL UYGULAMA
TASARIMI VE GERÇEKLEŞTİRİLMESİ

Fatih DİNÇ tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Dr. Öğr. Üyesi Fatih KAYAALP

Düzce Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Fatih KAYAALP

Düzce Üniversitesi

Prof. Dr. Resul KARA

Düzce Üniversitesi

Dr. Öğr. Üyesi Şafak KAYIKÇI

Abant İzzet Baysal Üniversitesi

Tez Savunma Tarihi: 07/12/2018

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarımı ihlal edici bir davranışımın olmadığını beyan ederim.

7 Aralık 2018

Fatih DİNÇ

TEŐEKKÜR

Yüksek Lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Dr. Öğr. Üyesi Fatih KAYAALP'e en içten dileklerle teşekkür ederim.

Çalışma sürecinde manevi desteğini her an yanımda hissettiğim sevgili eşim ve aileme minnettarım. Onlar olmadan, bunu başaramazdım.

Bu tez çalışması, Düzce Üniversitesi BAP- 2016.06.03.103 numaralı Bilimsel Araştırma Projesiyle desteklenmiştir.

7 Aralık 2018

Fatih DİNÇ

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	viii
KISALTMALAR	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
1.1. ALGORİTMA	2
1.1.1. Algoritma Tanımı	2
1.1.2. Algoritma Yapısı	2
1.1.3. Algoritmalarda Kullanılan Terimler	3
1.1.3.1. Tanımlayıcı	3
1.1.3.2. Değişken	3
1.1.3.3. Sabit	3
1.1.3.4. Atama	3
1.1.3.5. Döngü	4
1.1.4. Akış Şeması	4
1.2. UZAKTAN ÖĞRENME	5
1.3. ELEKTRONİK ÖĞRENME	6
1.4. MOBİL ÖĞRENME	6
1.5. E-ÖĞRENME İLE M-ÖĞRENME ARASINDAKİ FARKLAR	7
1.6. MOBİL ÖĞRENMENİN GETİRDİĞİ AVANTAJLAR	11
1.6.1. Yaşam Boyu Öğrenme	11
1.6.2. İhtiyaç Anında Öğrenme	12
1.6.3. Kendi İsteddiği Zaman ve Mekânda Öğrenme	12
1.6.4. Farkında Olmadan Öğrenme	12
1.6.5. M-Öğrenmenin Dezavantaj ve Kısıtlılıkları	14
1.7. UYGULAMA GELİŞTİRMEDE KULLANILAN MOBİL PLATFORMLAR	17
1.7.1. Yerel (Native) Platform	17
1.7.1.1. Native Uygulama Geliştirmenin Avantaj ve Dezavantajları	18
1.7.2. Hibrit (Hybrid) Platform	18
1.7.2.1. Hibrit Uygulama Geliştirmenin Avantaj ve Dezavantajları	19
1.7.3. Çapraz (Cross) Platform	19
1.7.3.1. Cross Uygulama Geliştirmenin Avantaj ve Dezavantajları	20
1.7.4. Uyumlu (Responsive) Platformlar	20
1.7.4.1. Responsive Geliştirmenin Avantaj ve Dezavantajları	20
2. MATERYAL VE YÖNTEM	23
2.1. IOS TARİHÇESİ	23
2.2. XCODE	27
2.3. VERSİYON KONTROL SİSTEMİ VE KULLANIMININ FAYDALARI	29

2.3.1. Yerel Versiyon Kontrol Sistemi.....	30
2.3.2. Merkezi Versiyon Kontrol Sistemi.....	30
2.3.3. Dağıtık Versiyon Kontrol Sistemi	31
2.4. CORE DATA	32
2.5. SKETCH İLE TASARIM.....	34
2.6. ALGORİTMA EĞİTİMİ İÇİN UYGULAMANIN GELİŞTİRİLMESİ	36
2.6.1. Veritabanı.....	36
2.6.2. Uygulamanın Kullanımı	37
2.7. KULLANICILARIN UYGULAMAYI TEST EDEBİLMESİ İÇİN TESTFLIGHT KULLANILMASI	42
2.8. UYGULAMA AŞAMASI.....	46
3. SONUÇLAR VE ÖNERİLER	48
3.1. SONUÇLAR	48
3.2. ÖNERİLER.....	49
4. KAYNAKLAR	51
ÖZGEÇMİŞ	55

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. M-öğrenmenin, e-öğrenme ve uzaktan öğrenmedeki yeri [23].....	8
Şekil 1.2. E-öğrenme ile m-öğrenme arasındaki ilişki [25].....	8
Şekil 1.3. Uzaktan eğitim ve mobil öğrenme [35].....	13
Şekil 2.1. IOS kullanan cihazların dağılımı [56].....	26
Şekil 2.2. Xcode arayüzü.....	28
Şekil 2.3. Örnek playground görüntüsü.....	29
Şekil 2.4. Yerel versiyon kontrol sistemi diyagramı [57].....	30
Şekil 2.5. Merkezi versiyon kontrol diyagramı [57].....	31
Şekil 2.6. Dağıtık versiyon kontrol diyagramı [57].....	32
Şekil 2.7. Sketch çalışma yüzeyi ekleme.....	35
Şekil 2.8. Core data tabloları.....	36
Şekil 2.9. Uygulama ekranları a) Başlama ekranı b) Ana menü.....	37
Şekil 2.10. Konu anlatımı a) Menü b) İçerik.....	38
Şekil 2.11. Çözümlü testler a) Test menüsü b) Soru adedi.....	39
Şekil 2.12. Konu testleri a) Çoktan seçmeli b) Doğru/Yanlış.....	39
Şekil 2.13. Konu testi cevap anahtarı.....	40
Şekil 2.14. Akış diyagramı soru ekranı.....	41
Şekil 2.15. Akış diyagramı cevap anahtarı a) Doğru b) Boş c) Yanlış.....	41
Şekil 2.16. Simülatör seçimi.....	43
Şekil 2.17. Yükleme paketi oluşturma.....	43
Şekil 2.18. Yükleme seçenekleri.....	43
Şekil 2.19. Yükleme öncesi bilgilendirme.....	44
Şekil 2.20. Yükleme başarılı.....	44
Şekil 2.21. iTunesConnect hesabı ana menü.....	45
Şekil 2.22. Yüklenen uygulama.....	45
Şekil 2.23. TestFlight menüsü.....	45

ÇİZELGE LİSTESİ

	<u>Sayfa No</u>
Çizelge 1.1. Akış şemasında kullanılan şekiller.	5
Çizelge 1.2. M-öğrenme ortamındaki çeşitli değişiklikler [27].	9
Çizelge 1.3. E-öğrenme ile m-öğrenme terimlerinin karşılaştırılmasının [28].	11
Çizelge 1.4. Yerel platformların kullandığı diller.	18
Çizelge 1.5. Platformların avantajları ve dezavantajları.	21
Çizelge 2.1. IOS versiyon ve satışa sunulan cihaz bilgileri.	23
Çizelge 2.2. IOS 12 öncesi, versiyonların mobil cihazlarda kullanım oranı [55].	25
Çizelge 2.3. IOS 12 sonrası, versiyonların mobil cihazlarda kullanım oranı [56].	25
Çizelge 2.4. iPhone kullanıcılarının IOS kullanım dağılımı [56].	27
Çizelge 2.5. Veritabanı ile Core Data'nın karşılaştırılması [60].	34
Çizelge 2.6. Tablolar ile tutulan bilgiler.	37
Çizelge 2.7. Çalışma gruplarının başarı durumu.	47

KISALTMALAR

E-öğrenme	Elektronik Öğrenme
ICT	Information and Communication Technology
IDE	Integrated Development Environment
IOS	iPhone Operating System
M-öğrenme	Mobil Öğrenme
MacOS	Macintosh Operating System
MB	Megabyte
OS	Operating System
PDA	Personal Data Assistant
SDK	Software Development Kit
SQL	Structured Query Language
TÜİK	Türkiye İstatistik Kurumu
UI	User Interface
UX	User Experience
VKS	Versiyon Kontrol Sistemi
WWDC	Worldwide Developers Conference

ÖZET

ALGORİTMA EĞİTİMİ İÇİN BİR MOBİL UYGULAMA TASARIMI VE GERÇEKLEŞTİRİLMESİ

Fatih DİNÇ

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Fatih KAYAALP

Aralık 2018, 54 sayfa

Teknolojik gelişmeler sayesinde mobil cihaz kullanımı yaygınlaştı. Bu kullanım oranlarının artmasıyla ortaya çıkan mobil öğrenme, kullanıcıların internetteki kaynaklara, zaman ve mekândan bağımsız olarak mobil internet veya kablosuz bağlantılar aracılığıyla erişmelerine olanak tanımaktadır. Algoritma kelimesi yaygın olarak, bilgisayar mühendisliği bilim dalı içerisinde bulunan programlama dilleri ile bir yazılım yapmanın aşamalarını tanımlamak için kullanılmaktadır. Ancak, belirli bir problemin bir plan dahilinde çözüm yolunun belirlenmesi işlemine de algoritma ismi verilebilir. Bu bakış açısıyla aslında tüm mühendislik dalları içerisinde de geçmektedir. Bu nedenle algoritma eğitimi gerek bilgisayar mühendisliği gerekse de diğer mühendislik dalları için temel ve çok önemli bir konudur. Algoritma mantığını tam olarak oturtamayan bir mühendisin, meslek hayatı içerisinde bir problemin çözüm aşamalarını sağlıklı bir şekilde planlayıp gerçekleştirmesi ve sonuca ulaştırması çok zor olmaktadır. Algoritmanın önemi ve birçok kullanıcının, mobil cihazlarının her yerde yanlarında olması nedeniyle, IOS platformu için algoritma eğitimini desteklemek amacıyla bir uygulama geliştirme fikri ortaya çıkmıştır. Bu çalışma ile temel algoritma eğitiminde kullanılmak üzere IOS ile çalışan mobil cihaz kullanan kişilere yönelik bir uygulama geliştirilmesi amaçlanmıştır. Uygulama yerel platform kullanılarak geliştirilmiştir. Uygulamada temel algoritma konularının anlatıldığı ve soruların bulunduğu iki ana menü bulunmaktadır. Sorular ise çoktan seçmeli, doğru/yanlış ve akış diyagramı oluşturma olmak üzere üç tipte tasarlanmıştır. Kullanıcıların bu uygulama sayesinde bir akış diyagramını baştan sona oluşturabilmesi sağlanmıştır. Bu diyagramları, sürükle-bırak yöntemi ile oluşturabilmektedirler. Veritabanı içerisinde tanımlı soru ve cevaplara göre kendi yaptığı çözümleri kontrol ederek doğru ve yanlışlarını görebilme imkânları da bulunmaktadır. Bilgisayar mühendisliği lisans öğrencilerine kullanılan uygulamanın, öğrencilerin başarısına olumlu etki yaptığı görülmüştür.

Anahtar sözcükler: Algoritma, Akış diyagramı, IOS, Mobil, Mobil öğrenme.

ABSTRACT

A MOBILE APPLICATION DESIGN AND IMPLEMENTATION FOR ALGORITHM EDUCATION

Fatih DİNÇ

Düzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-Electronics
and Computer Engineering

Master's Thesis

Supervisor: Assist. Prof. Dr. Fatih KAYAALP

December 2018, 54 pages

Thanks to technological developments, mobile device usage became widespread. Mobile learning, resulting from the increase of usage rates, allows users to access the resources on the Internet via mobile internet or wireless connections, regardless of time and place. Algorithm is commonly used to describe the stages of software written by programming languages within the computer engineering discipline. However, the process of determining a solution of a particular problem within a plan can be also called the algorithm. From this point of view, it is actually within all engineering branches. Because of this, algorithm education is a very basic and important subject for both computer engineering and other engineering disciplines. It is very difficult for an engineer who cannot fully understand the logic of the algorithm to plan achieve and realize the solution phases of a problem in his/her professional life. Because the importance of the algorithm and the fact that many users have mobile devices with them all the time, the idea of developing an application to support algorithm training for this platform has emerged. In this study, it is aimed to develop an application for people using mobile devices working with IOS for use in basic algorithm training. The application was developed using the native platform. There are two main menus in which the basic algorithm topics are explained and questions are existed. The questions are designed in three types: multiple choice, true/false and flow diagram creation. This application allows users to create a flow diagram from scratch. They can create the diagrams by drag-and-drop method. There is an opportunity to see the right and wrong answers by checking their own solutions according to the questions answers defined in the database. It has been observed that the application used by computer engineering undergraduate students has a positive effect on the success of the students.

Keywords: Algorithm, Flow diagram, IOS, Mobile, Mobile learning.

1. GİRİŞ

Günümüzde telefonlar yaşamımızla o kadar bütünleşti ki onlar yanımızda olmadığı zaman kendimizi huzursuz hisseder olduk. Öyle ki, sabah uyanan insanların birçoğunun yataktan kalkmadan önce mutlaka telefonlarını kontrol ettikleri bilinmektedir.

Akıllı telefonlar sadece iletişim kurmak amacıyla değil aynı zamanda öğrenme, kazanç işlem yapma kolaylığı ve eğlence amaçlı da kullanılabilen çok fonksiyonlu cihazlardır. Bütün bunlar da mobil uygulamaların gelişimini teşvik etmektedir.

Akıllı telefonların kullanımının yaygınlaşması, bilgisayar kullanımında azalmaya yol açmıştır. Hem taşınabilirlik hem de erişilebilirlik kolaylığı bu yaygınlaşmada etkili olmuştur. Yaklaşık 10 yıl öncesine kadar telefonların işlevi mesaj atmak ve arama yapmak ile sınırlı iken bugün bilgisayar üzerinden yapabileceğimiz internet üzerinden alışveriş, bankacılık işlemleri, sağlık hizmetleri, eğitim vb. ihtiyaçlarımızı akıllı telefonlar ile de yapabilir hale geldik. Bu sebeple de akıllı telefonlar hayatımızın ayrılmaz bir parçası haline geldi.

Türkiye İstatistik Kurumunun (TÜİK) yaptığı araştırmaya göre kullanıcıların 2016 yılı Nisan ayında %96.9'unda cep telefonu veya akıllı telefon bulunurken, sabit telefon bulunma oranı %25.6'dır. Ayrıca kullanıcıların evlerinde %22.9'unda masaüstü bilgisayar, %36.4'ünde taşınabilir mevcut iken tablet bilgisayar bulunma oranı %29.6'dır. 2015 yılında %20.9 olan internete bağlanabilen TV oranı ise 2016 yılında %24.6 olarak hesaplanmıştır [1], [2]. İnternet kullanan bireylerin oranı 2016 yılında %61.2 iken 2018 yılında bu oran %72.9'a yükselmiştir [1], [3].

Mobil cihazlarda iPhone Operating System (IOS), Android, Symbian, Windows Mobile, Blackberry OS gibi işletim sistemleri kullanılmaktadır. Yaygın olarak ise 3 işletim sistemi kullanılmaktadır. Bunlar Apple firmasının piyasaya sürdüğü IOS, Google firmasının piyasaya sürdüğü Android ve Microsoft firmasının piyasaya sürdüğü Windows Mobile'dır.

Günümüzde mobil cihazlar için farklı amaçlar için geliştirilen birçok uygulama bulunmaktadır. Eğitim, sağlık, eğlence, sosyal medya, haber gibi pek çok alanda tablet ve mobil cihazlar için uygulama geliştirilmektedir. 2015'te dünyadaki 1 milyar akıllı telefon pazarının %80.7'si Android'e, %17.7'si IOS'a aitti [4]. IOS işletim sistemindeki

uygulamaların 2018 Eylül ayı itibariyle %8.5'ini eğitim uygulamaları oluşturmaktadır. Uygulama kategorileri içinde en çok uygulaması bulunan 3. kategoridir. Birinci sırada %24.86 ile oyun uygulamaları, ikinci sırada ise %9.77 ile iş uygulamaları yer almaktadır [5]. 2008 yılının Temmuz ayında 800 uygulama bulunduran App Store'da, Ocak 2017 itibariyle 2.2 milyon uygulama bulunmaktadır [6]. Android market ise 2009 yılının Kasım ayında 16000 uygulama bulundururken, 2017 Mart ayı itibarıyla 2.8 milyon uygulama bulundurmaktadır [7].

1.1. ALGORİTMA

1.1.1. Algoritma Tanımı

Bir bilgisayar programı, aslında sıralı olarak tanımlanmış bir dizi komuttan başka bir şey değildir. Bu açıdan bizim yazmaya çalışacağımız program da bir dizi komut yani eylem topluluğudur. Her programda bu eylemler yazıldıkları sırada gerçekleştirilir veya çalıştırılırlar. Aslında bizim günlük hayattaki yaşantı tarzımız dahi düzenli olarak birtakım işlemlerin sıra ile yapılması şeklindedir. Yani bir iş yapabilmek için bir takım alt iş veya olayları peş peşe gerçekleştiririz. Herhangi bir problemi çözebilmek için gerekli olan sıralı mantıksal adımların tamamına algoritma denmektedir. Algoritma tasarımında aşağıdaki ifadelerin doğrulanması gerekmektedir [8].

- Her adım son derece belirgin olmalıdır.
- Algoritma belirli sayıda adım sonunda sonlandırılmalıdır.
- Karşılaşılabilecek bütün ihtimalleri ele alabilecek kadar genel olarak tasarlanmalıdır.

1.1.2. Algoritma Yapısı

Algoritma yazımı, çözümün başladığını gösteren *Başla* ifadesi ile başlamakta ve işlemlerin bittiğini belirten *Bitir* ifadesi ile sona ermektedir.

Problemi çözmek için uygulanacak bir algoritmanın temel adımlarını 3 gruba ayırabiliriz;

- Atama adımı: Değişkenlere değer atanması.
- Aritmetik adım: Toplama, bölme, üs alma vb.
- Mantıksal adım: Değerlerin karşılaştırılması

1.1.3. Algoritmelerde Kullanılan Terimler

Algoritmelerde tanımlayıcı, değişken, sabit, atama, döngü gibi terimler kullanılmaktadır. Kullanılan bu terimleri inceleyelim [9].

1.1.3.1. Tanımlayıcı

Tanımlayıcı, programı yazan geliştirici tarafından oluşturulan ve programdaki sabitleri, değişkenleri, kayıt alanlarını, özel bilgi tiplerini vb. adlandırılmak için kullanılan kelimelerdir. Bu kelimeler oluşturulurken aşağıdaki kurallara uyulması gerekmektedir.

- İngiliz alfabesindeki harflerden a-z ve A-Z arası kullanılabilir.
- Rakamlar kullanılabilir.
- Simgelerden alt çizgi (_) kullanılabilir.
- İsimlendirme, harf veya alt çizgi ile başlayabilir.
- İsimlendirme, kullanılan programlama dilinin komutu veya saklı kelimelerinden olamaz.

1.1.3.2. Değişken

Program çalıştırıldığında, farklı değerler alabilen veya aktarabilen bilgi alanları değişken olarak isimlendirilir. Değişkenler, programcının isteğine bağlı olarak isimlendirilebilir. Bunu yaparken tanımlayıcı kurallarına uyulması gerekmektedir.

1.1.3.3. Sabit

Sabit, programdaki değeri değişmeyen tanımlamalara denilmektedir. Tanımlama kurallarına uygun olarak oluşturulan sabitlere, alfasayısal veriler tırnak içinde aktarılırken sayısal veriler ise doğrudan aktarılır.

1.1.3.4. Atama

Atama, bilgi alanına veri yazma, bir ifadenin sonucunu başka bir değişkende gösterme gibi görevlerde kullanılan operatördür.

$$\text{Değişken} = \text{İfade}$$

Yukarıdaki satırda 'Değişken' herhangi bir sabitin/değişkenin ismini, 'İfade' ise matematiksel veya alfasayısal değerleri temsil etmektedir. '=' sembolü atama operatörüdür ve sağdaki değeri soldaki değişkene atar.

1.1.3.5. Döngü

Programlarda bazı işlemler belirli sayıda yapılmakta ve belirli ardışık değerler ile gerçekleştirilmektedir. Döngü, programlardaki işlem bloklarını, belirtilen sayıda gerçekleştiren işlem akış çevrimleridir.

1.1.4. Akış Şeması







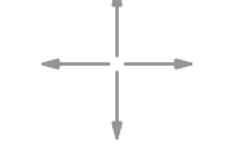

Algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş şekli akış şeması (flow chart) olarak adlandırılır. Akış şemalarının algoritmadan farkı, adımların simgeler halinde şekillerin içine yazılıyor olması ve adımlar arasındaki ilişkilerin ve yönünün oklar ile gösteriliyor olmasıdır.

Semboller en basit şekliyle kutulardan ve oklardan oluşur. Akış şemasının sembolleri American National Standards Institute (ANSI) tarafından belirlenmiştir ve tüm dünyada bu standarda uygun olarak kullanılmaktadır [10].

Akış semalarının hazırlanmasına; sorun çözümlenmesi sürecinin daha kolay anlaşılır biçime getirilmesi, iş akışının kontrol edilmesi ve programın kodlanmasının kolaylaştırılması gibi sebeplerle başvurulmaktadır. Uygulamalarda genellikle, yazılacak programlar için önce programın ana adımlarını gösteren genel bir bakış akış şeması hazırlanır. Daha sonra programın her adımı için ayrıntılı akış şemasının çizimi yapılır.

Yapılacak her iş veya komutlar simgelerle gösterilir. Çizelge 1.1’de verilen simgeler, akış şemalarının hazırlanmasında kullanılmaktadır.

Çizelge 1.1. Akış şemasında kullanılan şekiller.

 <p>Başla</p>	Başla: Algoritmanın başladığı konumu göstermektedir.
	Giriş/Okuma: Klavye aracılığıyla veri girişi yada okuma yapılacağını gösterir.
	İşlem/Atama: Aritmetik işlem yapımında ve değişkenlere değer atamada kullanılır.
	Karar: Algoritmada karar verme işleminde kullanılır.
	Döngü: Birden çok tekrar gerektiren durumlarda kullanılır.
	Çıktı: Hangi bilginin çıktısının alınacağı gösterir.
	Akış Çubuğu: Şekiller arasındaki akış yönünü gösterir.
	Bitir: Algoritmanın bittiği konumu göstermektedir.

1.2. UZAKTAN ÖĞRENME

Genellikle coğrafi olarak uzak olanların, öğrenmeye erişim sağlama çabasının anlatımında kullanılır. Yeni teknolojiler ortaya çıktıkça öğrenme, her türlü eğitimin odak noktası gibi görünmeye başladı ve uzaktan eğitimi bir kez daha mesafe, yani zaman ve yer ile ilgili sınırlamalarına odaklanmak için kullanıldı [11]. Bu terim daha sonra elektronik öğrenme, sanal öğrenme, web tabanlı öğrenme, çevrimiçi işbirlikçi öğrenme

vb. öğrenme biçimlerini açıklamak için geliştirilmiştir [12]. Etkileşim, öğrenmenin en önemli faktörlerindedir [13]. Uzaktan eğitim açısından etkileşim, geleneksel olarak üç kategoriye ayrılmıştır. Bunlar; içerikle etkileşim, öğretim elemanı ile etkileşim ve öğrencilerle etkileşimdir [14]. Sistemle etkileşim olarak adlandırılan dördüncü bir etkileşim kategorisi ise elektronik öğrenme kapsamında tanımlanmalıdır [15].

1.3. ELEKTRONİK ÖĞRENME

Elektronik öğrenme (e-öğrenme) teriminin kökenlerinin 1980'lerde ortaya çıktığı söylenmektedir, fakat bu kesin değildir [16]. E-öğrenme, sadece CD-ROM, internet veya intranet aracılığıyla sunulan içerik ve öğrenim yöntemlerini kapsamamaktadır [17]. E-öğrenme bilgiyi kullanıcılara ulaştıran, zaman kısıtlaması ya da yer yakınlığı gereksinimi olmayan web temelli çalışan sistemdir. E-öğrenme, eğitim veya çalışma etkinliklerinde bilgiyi ulaştırmak için telekomünikasyon imkanlarının kullanılmasıdır. Hızla gelişen teknoloji, e-öğrenmenin eğitimde yükselen bir değer haline gelmesini sağlamıştır [18]. E-öğrenme, bütün dersler için öğrenme zamanını seçme , öğrenim gören kişinin zamanla ilgili sınırlılığını aşma, soru sorma özgürlüğü, dersin dokümanlarına erişebilme, fikirlerini ifade edebilme gibi avantajları beraberinde getirmiştir [15]. E-öğrenme ortamlarını oluştururken dört faktörün düzenlenmesi gereklidir. Bunlar; ortam memnuniyeti, ortam karakteristiği, öğrenme etkinlikleri ve öğrenenlerin karakteristikleri olarak sıralanabilir [19].

1.4. MOBİL ÖĞRENME

Mobil öğrenme (m-öğrenme) için içerdiği karakteristik özelliklerine göre çok farklı tanımlar bulunmaktadır. Ancak zaman ve mekan sınırlaması olmaksızın eğitim içeriğine ve ortamına erişim olanağı olması ortak bir tanım olarak gösterilebilir [20].

Mobil uygulamaların esnekliğine dayanarak bazı üniversite ve okullar, müfredatlarına ve ihtiyaçlarına göre öğrencileri için özel uygulamalar geliştirdiler. Princeton üniversitesi 2010 yılında m-öğrenmeyi uygulamaya başladı. Böylece öğrenciler farklı formattaki dokümanlara erişebiliyorlar, duyuru yayımlayabiliyorlar, bloglara içerik yükleyebiliyorlar, derslere içerik yükleyebiliyorlar, test çözebiliyorlar ve önemli ders güncellemeleri ile değişiklikler hakkında bildirim alabiliyorlardı [21].

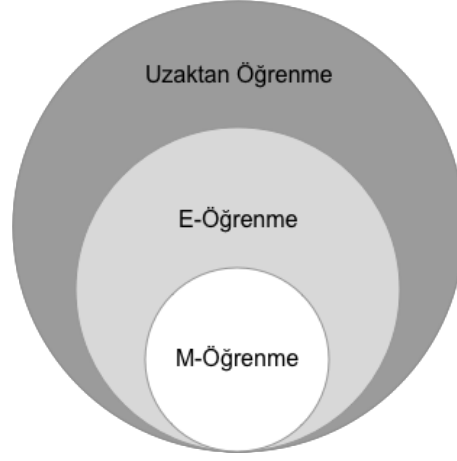
M-öğrenme ile ilgili birçok araştırma yapılmasına rağmen son araştırmalar, araştırmada daha iyi tasarım ve geliştirme aşaması üzerinde etkili olabilecek öncelikler bulunduğunu göstermektedir [22]. Bunlar;

- Öğretim ve öğrenim stratejileri
- Teminatlar
- Teori
- Öğrenmenin düzenlenmesi
- Ölçüm/Değerlendirme
- Öğrenenler
- Mobil teknoloji ve arayüz tasarımı
- Artırılmış gerçeklik
- Altyapı ve yönetim
- Dijital bölünme

1.5. E-ÖĞRENME İLE M-ÖĞRENME ARASINDAKİ FARKLAR

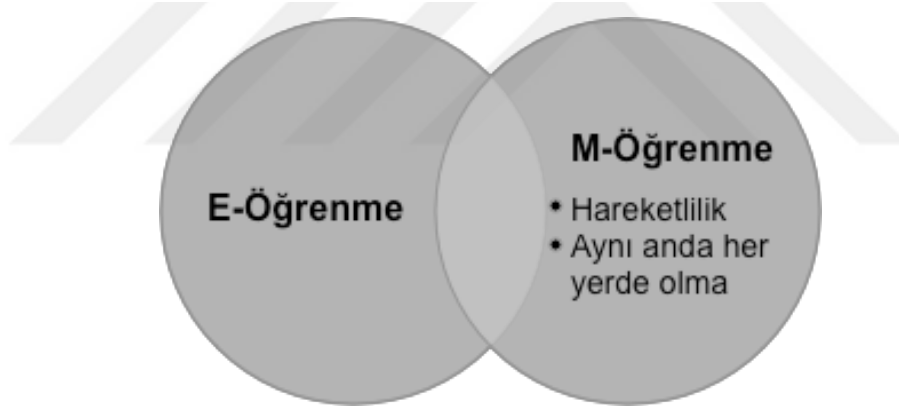
E-öğrenme internet teknolojileri aracılığıyla gerçekleştirilen web tabanlı eğitim, öğrenim ve bilgi yönetimi faaliyetleri olarak tarif edilmektedir. Ayrıca, görsel ve işitsel öğelerle zenginleştirilmiş eğitimsel içerikleri ve bunları destekleyen çeşitli testleri sunabilen, konu ile ilgili gerekli bilgilere ulaşımı kolaylaştırabilen ve en önemlisi interaktif ortam sağlayabilen bir öğrenme sistemi olarak da adlandırılmaktadır. M-öğrenme ise genellikle dijital öğrenmenin alt alanı olan e-öğrenme alanı içinde bir alt alan olarak tarif edilmektedir.

M-öğrenme ile e-öğrenmenin pek çok açıdan benzer yönleri vardır. Bu da bazen çalışmalarda birbiri yerine kullanılmasıyla sonuçlanabilmektedir. Georgiev ve arkadaşları 2004 yılında sundukları bir çalışmada, m-öğrenmeyi e-öğrenmenin bir bölümü olarak görmüş, dolayısıyla da uzaktan öğrenmenin bir bölümü olarak görmüşlerdir [23]. Şekil 1.1’de bu öğrenme türlerinin aralarındaki ilişki gösterilmiştir.



Şekil 1.1. M-öğrenmenin, e-öğrenme ve uzaktan öğrenmedeki yeri [23].

Peng ve arkadaşları, m-öğrenmeyi e-öğrenmeden ayıran iki özelliğin mobil/hareketli olma ve aynı anda her yerde olma olduğunu söylemektedirler. Mobil olmanın kendisinin bile öğrenmeye ilişkin yöntemleri değiştirmeye yeteceğini ve en ufak zaman dilimlerini bile öğrenme fırsatı olarak değerlendirmeyi sağlayacağını savunmaktadırlar [24]. Şekil 1.2’de m-öğrenmenin e-öğrenmeden farkı gösterilmiştir.



Şekil 1.2. E-öğrenme ile m-öğrenme arasındaki ilişki [25].

Öğrenmeyi daha yaygın ve erişilebilir hale getirme gücüne sahip oldukları için, mobil cihazlar, çoğu kişi tarafından e-öğrenimin doğal bir uzantısı olarak görülmektedir [26].

Çizelge 1.2’de Mehdipour ve Zerehkafi’ye göre, m-öğrenmenin e-öğrenmede meydana getirdiği değişiklikler bulunmaktadır [27].

Çizelge 1.2. M-öğrenme ortamındaki çeşitli değişiklikler [27].

Konu	E-Öğrenme	M-Öğrenme
Yer	Derslikler veya bilgisayar laboratuvarlarında	Her zaman ve her yerde
Pedagojik değişim	Metin ve grafik tabanlı talimatlar	Ses, grafik ve animasyon tabanlı talimatlar
	Derslikler veya bilgisayar laboratuvarlarında	Öğrenme alanında veya mobil öğrenme
Eğitimci – Öğrenci İletişimi	Pasif iletişim	Anında iletişim
	Eşzamansız	Eşzamanlı
	Önceden zamanlanmış	Spontan
Öğrenci – Eğitimci İletişimi	Yüz yüze	Esnek
	Genellikle sesli görüşme	Sesli ve görüntülü görüşme yapılabilir
	E-mail	7/24 anlık mesajlaşma
	Belirli yerlerde	Mekân sınırı yok
	Grup bilinci nedeniyle zayıf iletişim	Birebir iletişim, azaltılmış kısıtlamalar sayesinde zengin iletişim
Öğrenciye geri bildirim	Bire bir mümkün	Bire bir mümkün
	Asenkron ve gecikmeli	Senkron ve asenkron
	Standart eğitim	Özelleştirilebilir eğitim
	Kıyasa dayalı derecelendirme	Performansa ve gelişmeye dayalı derecelendirme
	Simülasyonlar ve laboratuvara dayalı deneyler	Gerçek hayat ve saha deneyleri
	Kâğıda dayalı	Daha az kâğıt, daha düşük maliyet

Çizelge 1.2. (devam). M-öğrenme ortamındaki çeşitli değişiklikler [27].

Konu	E-Öğrenme	M-Öğrenme
Ödevler ve Testler	Sınıfta veya bilgisayarda	Mekân sınırı yok
	Sınırlandırılmış süre	7/24 yapılabilir
	Kısıtlı süre	Süre sınırı olmadan yapılabilir
	Standart test	Bireyselleştirilmiş testler
	Genellikler gecikmeli geri bildirim	Anlık geri bildirim mümkün
	Sabit uzunlukta testler	Esnek uzunlukta/adette sorular
Sunumlar, Sınavlar ve Ödevler	Teorik ve metin tabanlı	Sahada pratiğe yönelik sınavlar
	Laboratuvarda gözlemlene ve izleme	Sahada gözlem ve uzaktan izleme
	Sınıf tabanlı sunumlar	Daha zengin iletişim ile bira bir sunumlar
	Genellikle tek dil kullanılır	Talimatların birçok dilde teslim edilmesi için otomatik çeviri (mümkün).
	Çoğunlukla bireyselleştirilmiş, bileşen tabanlı grup çalışması	Eşzamanlı ortak çalışma gurubu çalışması
	Kâğıt tabanlı ödev teslimatı	Elektronik tabanlı ödev teslimatı
	Belirli bir yerde ve zamanda ödevlerin elektronik olarak teslim edilmesi	Herhangi bir yerde ve zamanda ödevlerin elektronik olarak teslim edilmesi
	Eğitmenler zamanı ders vermek için kullanır	Eğitmenler zamanı bireysel talimat ve yardım için kullanır

Laouris ve Eteokleous'un, e-öğrenme ve m-öğrenme türleri arasında terimler açısından yaptığı karşılaştırma Çizelge 1.3'de gösterilmiştir [28].

Çizelge 1.3. E-öğrenme ile m-öğrenme terimlerinin karşılaştırılmasının [28].

E-Öğrenme	M-Öğrenme
Bilgisayar	Mobil
Bant genişliği	Gprs, 3G, Bluetooth
Multimedya	Nesneler
Etkileşimli	Spontane
Medya açısından zengin	Medya içeriği hafifletilmiş
Uzaktan öğrenme	Yerinde öğrenme
Daha resmi	Esnek
Benzetilme yapılmış durum	Gerçekçi durum
Hiper öğrenme	Yapılandırıcılık, durumculuk, işbirlikçi

1.6. MOBİL ÖĞRENMENİN GETİRDİĞİ AVANTAJLAR

Bireyin zaman ve mekân olarak özgür kalmasını, uzaktan eğitime talebin fazla olmasının en önemli nedenlerinden birisi olarak gösterilebilir. Öğrenmenin istenilen yer ve zamanda öğrenen tarafından başlatılıp, istediği anda sürece müdahale edebilmesi büyük bir özgürlüktür [29].

Mobil kablosuz teknolojilerin, eğitimde hareketliliğin sağladığı avantajları kullanması son zamanlarda trend haline geldi ve bugün yüksek öğrenimin en sıcak teknolojisi haline gelmektedir [30]. Eğitimin verimliliğini artıran mobil öğrenmenin, hareketliliğin sağladığı imkândan faydalandığı görülmektedir.

Öğrenen bireylere uzaktan eğitim; yaşam boyu öğrenme, farkında olmadan öğrenme, ihtiyaç anında öğrenme, kendi istediği zaman ve mekanda öğrenme sağlamaktadır [31].

Teknolojinin hızla gelişmesi, eğitim gibi yaşamın hemen her yönüne değinmiştir. Eğitimi, eğitimi desteklemek için kullanılan farklı birçok teknolojiyi hesaba katmadan düşünmek neredeyse imkansızdır [32].

1.6.1. Yaşam Boyu Öğrenme

Yaşam boyu öğrenme daha önce ortaya atılan kavramlardan farklı olarak, bireyi merkez alan bir yaklaşımın benimsenmesi, devletin eğitimdeki ağırlığının azaltılması, okul

rolünün deęiştirilmesi, karşılık olarak ise tarafların rollerinin güçlendirilmesi ve eğitimin belli bir zaman diliminde sınırlandırılmaması gerektiğinin vurgulanmasını savunmaktadır [33].

Öğrenen uzaktan eğitimde aslında kendi kendine öğrenen bireydir. Uzaktan öğrenen aslında yalnızdır [34]. Bu nedenle öğrenenin yaşam boyu öğrenme becerisi geliştirmesi için bilgilere farklı yollardan ulaşması ve bilgileri kendi kendine anlamlandırabilmesi gerekmektedir.

Ülkemizde 2000’li yıllarda gündeme gelen yaşam boyu öğrenmeye verilen önem, MEB’e bağlı olarak “Hayat Boyu Öğrenme” adında bir genel müdürlük kurulmasından anlaşılabilir.

1.6.2. İhtiyaç Anında Öğrenme

Bilgilere ihtiyaç duyulmadığı zaman önem verilmemekte ve böylece kolay bir şekilde öğrenilememektedir. Halbuki öğrenilecek bilgilere ihtiyaç anında ulaşılabilirse, hem faydası daha fazla olacak hem de anlık bilginin öğrenilmesi daha kolay ve kalıcı olacaktır.

1.6.3. Kendi İsteddiği Zaman ve Mekânda Öğrenme

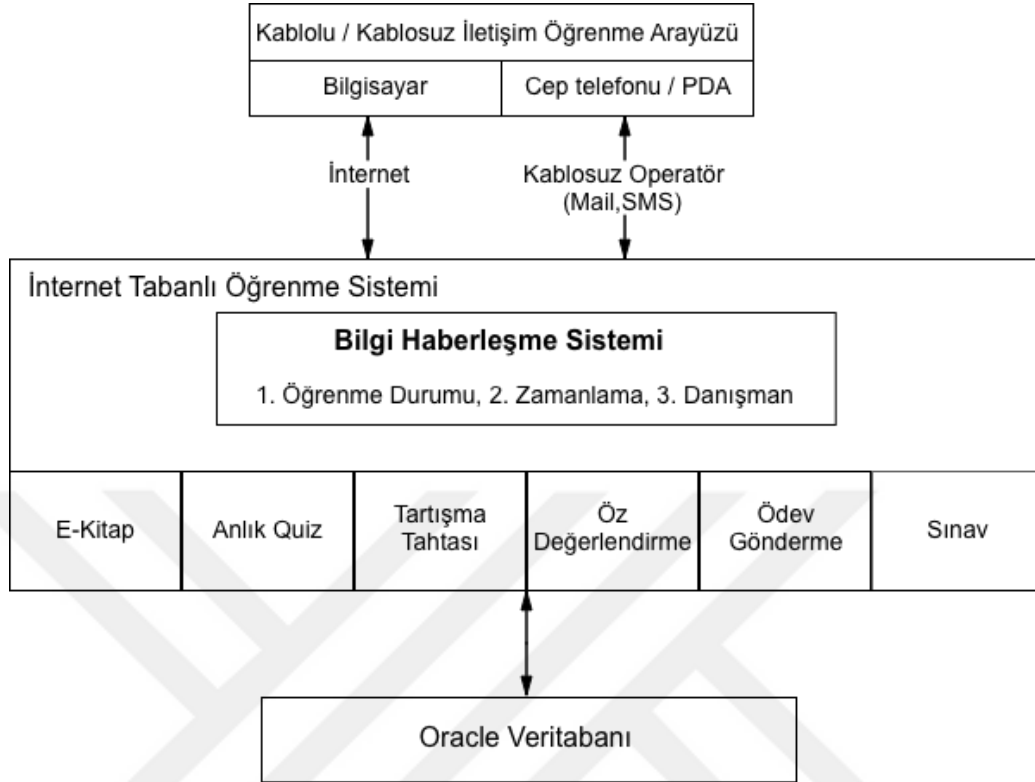
Bilgisayarlar ile yapılan uzaktan eğitim çalışmaları, zaman ve mekân bağımsız olarak adlandırılrsa da tam olarak gerçeği yansıtmamaktadır. Çünkü hem bilgisayarları her yere taşımak mümkün olmamakta, hem de bağlanılan internette kısıtlamalarla karşılaşmaktadır. Bu sebeplerden dolayı tam olarak zaman ve mekân bağımsız öğrenme mobil cihazlar ile mümkün olabilmektedir.

1.6.4. Farkında Olmadan Öğrenme

Sürekli öğretim çabalarının öğreneni bunaltması durumları sık sık ortaya çıkmaktadır. Bu sebeple çözüm olarak farkında olmadan öğrenme kavramı kullanılabilir. Öğrenenler algıda seçicilik, güdülenme eksikliği ya da dikkatin dağılması gibi nedenlerden dolayı birçok bilgiyi farkında olmadan öğrenebilmektedirler. Öğrenene sanki yaşamın bir parçasıymış gibi gerekli mesajlar ulaştırılarak yaşam boyu öğrenmenin gerekleri yerine getirilebilmekte ve öğrenme ortamının sıkıcı şartlarından arınarak farkında olmadan gerekli bilgiler öğretilmektedir. Günümüzdeki teknolojilerle bunu sağlamak büyük oranda kolaylaştırılmıştır [29].

Şekil 1.3’de verilen modelde öğrenen kişiye, mekâna ve zamana bağımlı olmadan derse ve bilgiye ulaşma imkânı sunulmakta, ayrıca eğitim eşitliği de mümkün kılınmaktadır.

Bu yapının oluşturulması ile kendi öğrenme sürecine müdahale edebilmesi, öğrenen kişiye çok büyük bir özgürlük sunmaktadır.



Şekil 1.3. Uzaktan eğitim ve mobil öğrenme [35].

Bu model sayesinde öğrenen kişi bilgisayar, cep bilgisayarları (PDA- personal data assistant), cep telefonu vb. araçlarla verilere ulaşabilmektedir. Ayrıca öğreneceği dersler ile ilgili bilgileri edinebilir ve değerlendirebilir, sınav alabilir, yaptığı işlemler veritabanında tutulduğu için öğrenen ile ilgili değerlendirmeler yapılabilir [35].

Mobil cihazlar çevrimiçi ve çevrimdışı olarak kullanabilmektedir. Her iki kullanımın da kendine göre avantaj ve dezavantajları bulunmaktadır. Bulun ve arkadaşları bunları şöyle açıklamıştır [29]:

Çevrimiçi kullanımın önemli avantajları bulunmaktadır. Güncellik, teorik olarak sınırsız bilgi ve senkron eğitim imkânı olarak sıralanabilmektedir. En önemli konulardan olan fayda/maliyet analizini yaparak, ihtiyaca göre imkânı kullanmayı düşünmek gerekmektedir. Gelecek yıllarda sürekli çevrimiçi kalmanın maliyetinin çok düşük olması beklenmektedir.

Çevrimdışı eğitimin de avantajları bulunmaktadır. Bunlar; maliyet, hız ve daha fazla mekân bağımsızlığı sağlamasıdır. Mekân bağımsızlığının daha fazla olması, kapsama alanı gibi bir problemin olmamasıyla sağlanmaktadır. Bir bilgiye ulaşmak istenildiğinde,

bilgiler doğrudan cihazda bulunduğundan, internet erişiminden doğabilecek bir gecikmeyle karşılaşılmayacak ve çok hızlı erişim sağlanacaktır. Ekstra bir bağlantı söz konusu olmadığı için, bağlantı ücreti de olmamaktadır.

1.6.5. M-Öğrenmenin Dezavantaj ve Kısıtlılıkları

M-Öğrenme teknolojinin gelişimine bağlı olarak, öncelikle iletişim ve eğlence amaçlı olarak kullanılması sebebiyle, içerisinde taşıdığı büyük eğitimsel potansiyelinin doğru yönde kullanılmaması nedeniyle birtakım dezavantaj ve riskleri de içerisinde bulundurmaktadır. Her ne kadar geleceğe bakıyor olsak da m-öğrenmenin sadece potansiyel faydalarına odaklanmak akılsızca bir davranış olacağı görülmektedir. [36]. M-öğrenmeyi ele alırken sihirli bir öğrenme fırsatı olarak görmek ve bu yönde bir tablo çizmek uygun bir yaklaşım olmamaktadır.

Öğrencilerin cep telefonlarından yaptıkları işlemler değerlendirildiğinde, en fazla sesli görüşme, fotoğraf çekme, video çekme, müzik dinleme, twitter ve facebook gibi sosyal ve kişisel amaçlarla kullandıkları ve büyük kısmının ise nadiren ders çalışma amaçlı kullandıkları görülmüştür [37].

Parsons, mobil öğrenmenin gelecekte oluşturabileceği önemli risklerden bahsetmiştir. Yerleşik dijital bölünmeler, dijital dikkat dağıtıcılar ve tehditler, yeşil manifestoya karşıtlık, sonuçlar üzerinde kontrol edilmeyen ve yanıltıcı etkiler oluşabilecek riskler olarak belirtilmektedir [36].

- Yerleşik dijital bölünmeler: Ülkeler arasındaki gelişmişlik farklarından dolayı, internet, mobil cihazlar gibi bilgi ve iletişim teknolojisinin (ICT) gelişmesiyle farklılıklar daha da artarak dijital bölünmeye neden olmuştur [38]. ICT'ye erişimde ayrıca bireyin becerileri, kalp rahatsızlığı, görme problemleri gibi sağlık problemleri de dijital bölünmeye sebep olmaktadır.
- Dikkat dağıtıcılar ve tehditler: Mobil cihazlar eğitim planı iyi bir şekilde yapılmadan kullanıma sokulursa, bu durumda öğrenme üzerinde olumlu bir uygulama yapılmak istenirken daha çok olumsuz bir etkiyle karşılaşılabilir. Öğrencilerin mobil cihazları, dersleriyle ilgilenmek yerine, mesajlaşma, facebook ve twitter gibi sosyal medya hesaplarını kontrol etmek için kullanmaları dikkatlerinin dağılmasına sebep olmaktadır. Ayrıca cihazların çalınması ve siber zorbalık konusunda da korkular bulunmaktadır [36]. Dikkat dağıtmanın bir başka boyutu da aşırı bilgi yüklenmesi potansiyelidir ve bireyi

öğrenme hedefinden uzaklaştırmaktadır. Amaç veri toplamak değil gerekli bilgiye ulaşmaktır [39].

- Yeşil manifestoya karşıtlık: Çek Cumhuriyeti’nde Polák ve Drápalová’nın 32.566 adet cep telefonu ile yaptığı bir araştırmada, cep telefonlarının ortalama kullanım süresinin 3.63 yıl, ortalama depolama veya yeniden kullanma süresinin 4.35 yıl olduğunun tespit edildiği görülmektedir [40]. Elektronik malzemenin, geri dönüşümü yapılan yerlerde bile, tehlikeli geri dönüşüm uygulamaları kişileri ve çevreyi zehirleyebilmektedir. Gelişmekte olan ülkeler üzerindeki etkiler ise ciddi sorunlara neden olabilmektedir [41]. Araştırmalardan hızla tüketilen dijital cihazların, gelecekte dünyada ciddi miktarda elektronik çöpü meydana getirebileceği anlaşılmaktadır. E-çöplerin oluşturduğu bu riski hükümetlerin dikkate alması gerekmektedir.
- Sonuçlar üzerinde kontrol edilmeyen ve yanıltıcı etkiler: 1972 yılında Dr. Fox ‘Hekim Eğitiminde Uygulanan Matematik Oyun Teorisi’ konulu bir bildiriye uydurma sözcükler ve karmaşık bir sunumla maskeleyerek anlatmıştı [42]. Dinleyen uzmanlar daha sonra değerlendirme formunda, bildirininkendilerini düşünmeye ittiğini, hatta çoğu, Dr. Fox’un konuyu çok iyi ele aldığını ve yeterli örnek verdiğini söylemişler [43]. Böylece insanların iyi sunulmuş bir şeyin değerini, içeriğinin gerçek değerine bakmadan daha kaliteli olarak görmesi ‘Dr. Fox etkisi’ olarak isimlendirilmeye başlanmıştır. Bu araştırmadan anlaşılacağı gibi iyi hazırlanmış bir doğru olmayan içerikle öğreneni yanıltmak mümkün olmaktadır.

Mobil cihazların kullanım oranının önemli derecede artması sayesinde, araştırmacılar m-öğrenmenin eğitim üzerinde önemli bir etkisi olduğunu öngörmektedirler [21]. Fakat, m-öğrenme uygulamasının tasarım ve geliştirilmesinin karmaşık bir süreç olduğu düşünülmektedir.

Danado & Paternò, sezgisel mobil interaktif ortamlar aracılığıyla, teknolojinin entegrasyonunu destekleyebilen uygulama geliştirme ortamının nasıl tasarlanacağını belirlemenin zor bir süreç olduğunu belirtmektedirler. Ayrıca, ekran boyutları sınırlı olan mobil platformlardaki sınırlamaların, dokunmaya dayalı etkileşim kullanımını nedeniyle daha karmaşık bir hal aldığını belirtmektedirler [44].

Asya’da m-öğrenme teşebbüslerini inceleyen UNESCO, raporlarında mobil öğrenmenin uygulanması ve devam ettirilebilmesi süreçlerinde karşılaşılan zorluk ve engelleri ortaya koymaktadır [45]. İncelemelerin sonucunda belirledikleri problemleri şu şekilde

belirtmişlerdir:

- Sınırlı telekomünikasyon ve elektrik altyapısı: Bazı yerleşim yerlerinde altyapı sorunlarından ve yüksek maliyetlerden dolayı elektrik olmaması nedeniyle, cihazlarını şarj etmek için komşu köylere gitmek durumunda kaldıkları tespit edilmiştir.
- Cep telefonlarının eğitim aracı olarak kullanılmasındaki şüpheler: Bunun nedeni olarak insanların cep telefonlarını öğrenme yerine iletişim ve eğlence ile ilişkilendirmesi gösterilmiştir. Saygın ve yerel bir sivil toplum örgütünden olan Bunyad yardımıyla katılımcılara mobil öğrenmenin faydalarının anlatılması ile şüpheler giderilmiştir [46].
- Sosyo-kültürel kaygılar ve güvenlik: Dini ve kültürel kısıtlamalar nedeniyle aşırılık yanlısı grupların kadınlara cep telefonu verme fikrine karşı olmaları, projeye karşı kuşku oluşturmuştur.
- Öğrenme materyallerinin yerel olarak uyumsuzluğu: Eğitimciler, ingilizceyi videolardakiler kadar iyi konuşamamayı utanç verici bularak, derslere hazırlanmak için çok fazla zaman harcamak durumunda kalmışlardır.
- Teknolojinin eskimesi: Mobil teknolojinin hızla gelişmesi nedeniyle mobil cihazlar ve yazılımların da gelişmesiyle, mevcut cihazların modası geçmiş olacağı ve ekipmanların değiştirilmesi gerekeceği belirtilmiştir.

M-öğrenme uygulamalarının geliştirilmesi ve tasarımı, eğitimcilerin müfredatlarında kendi başlarına m-öğrenme uygulamasını kullanmalarını daha zor hale getirebilen bilişsel süreçlere ve teknik becerilere ihtiyaç duymaktadır [47].

Jisc e-öğrenme programı tarafından 2005-2009 yılları arasında yürütülen bir araştırmada, araştırmacılar öğrencinin e-öğrenmeyi kullanma ve mobil cihazları kullanma deneyimlerini netleştirmeye çalışmışlardır. Araştırmalarına göre, öğrenciler tarafından kullanılan yeni teknolojilerin ve kurs öğretmenleri tarafından sağlanan geleneksel eğitimin karışımı, çalışma biçimlerini değiştirdiği ve akademik personelin beklentileri ile öğrencilerin çalışma alışkanlıkları arasında uyumsuzluğa neden olduğu sonucuna varmışlardır [48].

Mobil geliştirme ortamında cihazların sınırlı kapasiteleri, hızlı bir şekilde teknoloji gelişimi, çeşitli protokol, standart ve ağ teknolojileri, birden fazla platform için (IOS, Android, Windows Phone gibi) ayrı ayrı yazılım geliştirilmesi, mobil kullanıcıların kişisel ihtiyaçlarının farklılığı gibi durumlar sorun oluşturabilmektedir [49].

Mobil uygulamaların, uygulama marketine yüklenebilmesi için yayınlanma ücreti ödenmesi de uygulamalarla ilgili bir diğer negatif durumdur. Google uygulamayı yayımlamak için 25 \$ talep ederken, Apple uygulamaları yüklemek için yurtdışında yıllık 99 \$, Türkiye’de ise 279 TL talep etmektedir.

Sonuç olarak, öğrenenlerin fazla miktarda bilgiye erişim sağladıkları durumlarda, mobil öğrenmeyi uygulayan eğitimciler öğrenme etkinliğinin çerçevesini belirlemeli, öğrenenlerin mobil teknolojinin sınırsız dünyasında kaybolmasının önüne geçmelidir. Buna ek olarak m-öğrenmeyi uygulamak isteyen eğitimciler elektronik ortamdaki hangi kaynakların doğru ve güvenilir olduğunu kontrol etmelidir. Öğrenenlere mobil okur/yazarlığıyla ilgili bilgilendirmeler yapılarak, onların da eriştikleri bilgiler arasında değerlendirme yapabilmeleri sağlanması m-öğrenmenin etkisini artırmaktadır.

1.7. UYGULAMA GELİŞTİRMEDE KULLANILAN MOBİL PLATFORMLAR

Geliştirilen mobil uygulamaların ihtiyaca cevap veren ve teknolojik altyapıya uygun platformlardan tercih edilmesi gerekmektedir. Mobil uygulama geliştirirken yerel (native), melez (hybrid), çapraz (cross) ve uyumlu (responsive) platformlar kullanılabilir.

1.7.1. Yerel (Native) Platform

Günümüzde Apple Store ve Google Play gibi mobil uygulama mağazalarında milyonlarca ücretli veya ücretsiz uygulamaya ulaşılabilmektedir. Popüler olan IOS ve Android gibi farklı ortamlar için, farklı programlama dilleri ve geliştirme paketleri bulunmaktadır. Bunların başında Swift, Java, Objective-C programlama dilleri gelmektedir. Yerel platformlarda o ortamın gerektirdiği programlama dilinin kullanılması gerekmektedir. Ayrıca, yerel platformlarda, her programlama dili birbirinden farklı özelliklere sahip olduğu için farklı uzmanlıklar gerektirmektedir [50]. Çizelge 1.4’de kullanılacak platform için hangi programlama dilinin öğrenilmesi gerektiği listelenmiştir.

Çizelge 1.4. Yerel platformların kullandığı diller.

Platform	Kullanılan diller
IOS	Swift, Objective-C
Android	Java, Kotlin
RIM	Java
Symbian	C, C++, Html5/ Css3/ JS, Python
Windows Mobile	.Net

1.7.1.1. Native Uygulama Geliştirmenin Avantaj ve Dezavantajları

Avantajları;

- Kullanıcı arayüzü konusunda iyi performans elde edilir.
- Uygulama, geliştirilen platformun mağazasında yer alabilir.
- Mobil cihazların yazılım ve donanımlarına tam erişim sağlanır.
- Performans gerektiren gelişmiş düzeyde uygulama geliştirilebilir.

Dezavantajları;

- Geliştirilen farklı mobil cihaz platformu için gerekli beceri ve yazılım dillerinin öğrenilmesi gerekir.
- Geliştirilmesi masraflıdır.
- Aynı uygulamanın her mobil platform için yeniden kodlanması gerekir.

1.7.2. Hibrit (Hybrid) Platform

Temel olarak 'write once, run everywhere' prensibine göre, yani bir kere yazılmış olan bir kodun birden fazla ortamda çalıştırabilmesi için geliştirilmiş bir mobil uygulama platformudur. Farklı cihaz ve çözünürlüklerde HTML5, CSS3 ve JavaScript birlikte rahatlıkla çalıştırılarak kullanılabilir. Farklı ortamlarda çalıştırılabildiğinden geliştirilme aşaması, zaman ve bütçe açısından daha az maliyetli olmaktadır. Genellikle gömülü tarayıcılar üzerinde çalışmaktadırlar. Cihaz donanımını verimli kullanamayan hibrit uygulamaları, grafik işleme ve karmaşık hesapların yapılması gibi performans gerektiren işlemlerde kullanmak uygun olmamaktadır. Bu sebeple günümüzde çok fazla donanım ve etkileşim ihtiyacı için yerel geliştirme tercih edilirken, temel olarak daha düşük donanım gerektiren ve veri yükü daha az olan uygulamalarda hibrit geliştirme tercih edilmektedir. Corona, Cordova, jQuery Mobile, Titanium, PhoneGap gibi birçok

geliştirme yazılımları hibrit uygulama yapmak için geliştiricinin ihtiyaçlarını karşılayabilmektedir [50].

1.7.2.1. Hibrit Uygulama Geliştirmenin Avantaj ve Dezavantajları

Avantajları;

- Desteklenen platformlarda kod tekrar yazılabilir.
- HTML, CSS ve JavaScript ile yapılmış ortamlarda çalıştırılabilir.
- Eklentiler yoluyla genişletilebilir.
- Geliştirilen uygulamalar, uygulama marketlerine yüklenilebilir.

Dezavantajları;

- Kullanıcının arayüzü kullanma performansı, kullanılan native uygulamalardan etkilenir.
- Kullanılacak her mobil cihaz özelliğine göre web görünümlü uygulamalar gerekir.
- Kullanılan her bir Javascript kütüphanesi farklı entegre geliştirme ortamı (IDE) gerektirir.
- İşletim sistemlerinin yeni sürümlerinde uyumsuzluk gösterebilir.

1.7.3. Çapraz (Cross) Platform

Çapraz platform uygulamalarının çalışma performansları, hibrit platformlarla karşılaştırıldıklarında çok yüksek olmaktadır. Geliştiriciler, Xamarin ve QT gibi birden fazla platformu destekleyen bir grafiksel arayüze sahip geliştirme araçlarını kullanabilmektedir. Geliştiriciler, C# ile paylaşılan bir kod kütüphanesi kullanarak yerel kullanıcı arayüzüne sahip IOS, Android, Windows ve MacOS da dahil olmak üzere birden fazla platforma kod yazmak ve paylaşmak için Xamarin'i kullanabilmektedir. Mayıs 2015'ten itibaren dünya çapında 120'den fazla ülkede Xamarin ürünleri kullanılmaktadır. 24 Şubat 2016'da Microsoft, Xamarin'i satın alarak kendi firmasına katmış ve Microsoft Build 2016'da Visual Studio kullanıcılarının ücretsiz olarak kullanabileceğini açıklamıştır [51]. QT ise C++ kullanmaktadır. C++ temelli olmasına rağmen farklı dillerle olan bağlantısı sayesinde PHP, Perl, Pascal, C#, Java, Python ve Ruby ile de kullanılabilir [52].

1.7.3.1. *Cross Uygulama Geliştirme'nin Avantaj ve Dezavantajları*

Avantajları;

- Birçok cihazın uygulamasına erişim mümkün olmaktadır.
- Geliştirilen uygulamaların, mobil cihazların uygulama marketlerine yüklenmesi mümkün olmaktadır.
- Desteklenen platformlar arasında kodlar tekrar kullanılabilir.

Dezavantajları;

- Hata ayıklama zor olabilir.
- Geliştirilen uygulama tüm ortamlarda desteklenmeyebilir.
- Native platformda yapılan güncellemelerle uyum problemi olabilir.

1.7.4. **Uyumlu (Responsive) Platformlar**

Web sayfalarının tasarımının farklı platformlara göre uyumunu sağlamaktadır. Responsive ile yapılmış bir web sitesi masaüstünde farklı görünürken mobil cihazlarda (tablet, akıllı telefon vb.) ve değişik ekran boyutlarında daha farklı görünebilmektedir.

Günümüzde birçok web tasarımcısı ve geliştiricisi responsive web tasarımı, mobil uygulamalara göre daha uygun maliyetli olması sebebiyle önermektedir ve tercih etmektedir. Örneğin; test.com'u cep telefonlarından ziyaret edebilmek için m.test.com adlı yeni bir mobil site oluşturmak gerekirken responsive tasarımda bu zorunluluk ortadan kalkmaktadır. Tasarımın daha kolay ve esnek bir şekilde yapılabilmesi için de kullanılmaktadır. Ayrıca fazla kod bilgisi ve becerisi gerektirmediği için de tercih sebebi olmaktadır.

JavaScript, HTML5, CSS gibi web teknolojileri kullanılarak geliştirme yapılabilmektedir. Apple ve Google marketlerine ek bir uygulama yükleme gerektirmeden yalnızca web tarayıcısıyla kullanılabilir [50]. Bu sayede çok büyük bir kullanıcı kitlesine ulaşılabilir.

1.7.4.1. *Responsive Geliştirme'nin Avantaj ve Dezavantajları*

Avantajları;

- Esnek tasarımı sayesinde, var olan web siteleri mobile uyarlanarak kullanımı sağlanmaktadır.
- Geniş kitlelere ulaşması sağlanmaktadır.

- Geliştirirken kullanılan kodları, platformlar arasında tekrar kullanmak mümkün olmaktadır.

Dezavantajları;

- Uygulama marketlerine yükleme yapılamaması nedeniyle kullanıcının keşfetme imkânı sınırlanmaktadır.
- Uygulama arayüzlerine erişimi sınırlanmaktadır.
- Kullanılan cihazın donanımsal özelliklerinin (kamera vb.) kullanılması mümkün olmamaktadır.

Çizelge 1.5’de mobil uygulama geliştirilirken kullanılan platformların avantaj ve dezavantajları tablo halinde verilmiştir.

Çizelge 1.5. Platformların avantajları ve dezavantajları.

Platform	Avantaj	Dezavantaj
Native Platform	<p>Kullanıcı arayüzü konusunda iyi performans elde edilir.</p> <p>Uygulama geliştirilen platformun mağazasında yer alabilir.</p> <p>Mobil cihazların yazılım ve donanımlarına tam erişim sağlanır.</p> <p>Performans gerektiren gelişmiş düzeyde uygulama geliştirilebilir.</p>	<p>Geliştirilen farklı mobil cihaz platformu için gerekli beceri ve yazılım dillerinin öğrenilmesi gerekir.</p> <p>Geliştirilmesi masraflıdır.</p> <p>Aynı uygulamanın her mobil platform için yeniden kodlanması gerekir.</p>
Hybrid Platform	<p>Desteklenen platformlarda kod tekrar yazılabilir.</p> <p>HTML, CSS ve JavaScript ile yapılmış ortamlarda çalıştırılabilir.</p> <p>Eklentiler yoluyla genişletilebilir.</p>	<p>Kullanıcının arayüzü kullanma performansı, kullanılan native uygulamalardan etkilenir.</p> <p>Kullanılacak her mobil cihaz özelliğine göre web görünümlü uygulamalar gerekir.</p> <p>Kullanılan her bir JavaScript kütüphanesi farklı entegre geliştirme ortamı (IDE) gerektirir</p>

Çizelge 1.5. (devam). Platformların avantajları ve dezavantajları.

Platform	Avantaj	Dezavantaj
Hybrid Platform	Geliştirilen uygulamalar, uygulama marketlerine yüklenilebilir.	İşletim sistemlerinin yeni sürümlerinde uyumsuzluk gösterebilir.
Cross Platform	Birçok cihazın uygulamasına erişim mümkün olmaktadır. Geliştirilen uygulamaların, mobil cihazların uygulama marketlerine yüklenmesi mümkün olmaktadır. Desteklenen platformlar arasında kodlar tekrar kullanılabilir.	Hata ayıklama zor olabilir. Geliştirilen uygulama tüm ortamlarda desteklenmeyebilir. Native platformda yapılan güncellemelerle uyum problemi olabilir.
Responsive Platform	Esnek tasarımı sayesinde, var olan web siteleri mobile uyarlanarak kullanımı sağlanmaktadır. Geniş kitlelere ulaşması sağlanmaktadır. Geliştirirken kullanılan kodları platformlar arasında tekrar kullanmak mümkündür.	Uygulama marketlerine yükleme yapılamaması nedeniyle kullanıcının keşfetme imkânı sınırlıdır. Uygulama arayüzlerine erişimi sınırlıdır. Kullanılan cihazın donanımsal özelliklerinin (kamera vb.) kullanılması mümkün olmamaktadır.

2. MATERYAL VE YÖNTEM

2.1. IOS TARİHÇESİ

İşletim sistemi ilk olarak iPhone için Macworld Conference & Expo'da 9 Ocak 2007'de duyuruldu, aynı yılın haziran ayında yayınlandı. İlk başta Apple pazarlama literatürü bu ismi açıkça belirtmedi, "iPhone OS X kullanıyor" diye açıkladı. Apple 7 Haziran 2010'da gerçekleştirilen Worldwide Developers Conference (WWDC)'da, iPhone OS'un adını IOS olarak değiştirdiğini açıkladı [53].

Başlangıçta, üçüncü tarafların geliştirdiği uygulamalar desteklenmiyordu. Steve Wozniak daha sonra geliştiricilerin uygulamalar geliştirmesini savundu. 17 Ekim 2007'de Apple SDK'in geliştirildiğini duyurdu ve geliştiriciler için Şubat ayında yayınladı.

IOS, Apple'ın başlangıçta iPhone için geliştirdiği ancak daha sonra iPod touch ve iPad'de de kullanılan mobil işletim sistemine verilen isimdir. MacOS'dan türetilmiştir. İşletim sistemi, cihazın içinde yaklaşık 2500 MB'lık bir alan kaplamaktadır. IOS sistemi yapısı nedeniyle Apple, AppStore ve iTunes dışında hiçbir yerden uygulama yüklenememektedir [54].

Çizelge 2.1'de IOS'un piyasaya sürülmüş versiyon bilgileri, beraberinde satışa çıkarılan cihazlar ve ekran boyutları görülmektedir.

Çizelge 2.1. IOS versiyon ve satışa sunulan cihaz bilgileri.

Versiyon	Birlikte Satışa Sunulan Cihaz	Sürüm Yılı	Ekran Boyutu
iPhone OS 1	iPhone (1. Nesil)	2007	3.5"
iPhone OS 2	iPhone 3G	2008	3.5"
iPhone OS 3	iPhone 3GS	2009	3.5"
IOS 4	iPhone 4	2010	3.5"
IOS 5	iPhone 4S	2011	3.5"
IOS 6	iPhone 5	2012	4"

Çizelge 2.1. (devam). IOS versiyon ve satışa sunulan cihaz bilgileri.

Versiyon	Birlikte Satışa Sunulan Cihaz	Sürüm Yılı	Ekran Boyutu
IOS 7	iPhone 5S	2013	4"
IOS 8	iPhone 6	2014	4.7"
	iPhone 6 Plus		5.5"
IOS 9	iPhone 6S	2015	4.7"
	iPhone 6S Plus		5.5"
IOS 10	iPhone SE	2016	4"
	iPhone 7		4.7"
	iPhone 7 Plus		5.5"
IOS 11	iPhone 8	2017	4.7"
	iPhone 8 Plus		5.5"
	iPhone X		5.8"
IOS 12	iPhone XS	2018	5.8"
	iPhone XS Max		6.5"
	iPhone XR		6.1"

Çizelge 2.1'den de görüldüğü üzere Apple firması her yıl ürettiği cihaz türlerini kısıtlı tutmuştur. Bu da hem kullanıcıya seçim yapma konusunda kolaylık sağlamış hem de geliştiriciye kolaylık sağlamıştır. Geliştirici Google firmasına ait Android işletim sistemindeki gibi sürekli farklı cihazların ekran boyutu için uyumlu tasarım yapma problemini minimuma indirmiştir.

Çizelge 2.2'de Nisan 2018'de kullanılan, Çizelge 2.3'de ise Kasım 2018'de kullanılan bütün iPhone, iPad ve iPod cihazlarının IOS sürümlerinin dünya genelinde kullanım oranları verilmiştir.

Çizelge 2.2. IOS 12 öncesi, versiyonların mobil cihazlarda kullanım oranı [55].

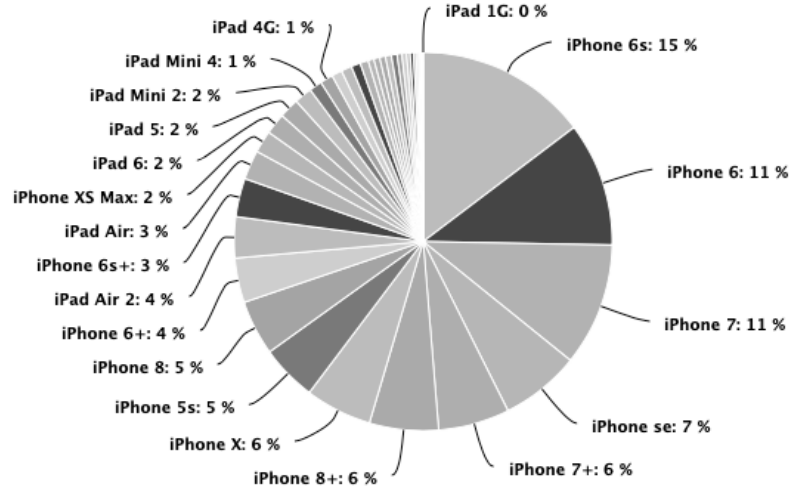
IOS Versiyon	Kullanım oranları
11.X	78.7%
10.X	10.9%
9.X	6.2%
8.X	0.7%
7.X	1.4%
6.X	1.9%
5.X	0.2%

Çizelge 2.3. IOS 12 sonrası, versiyonların mobil cihazlarda kullanım oranı [56].

IOS Versiyon	Kullanım oranları
12.X	%65.2
11.X	%27.1
10.X	%4
9.X	%2.9
8.X	%0.1
7.X	%0.2
6.X	%0.3
5.X	%0.1

Çizelge 2.2’de Nisan 2018’de Apple’ın en son piyasaya sürdüğü IOS 12 öncesi, IOS 11 kullanım oranının %78.7 [55] olduğu görülmektedir. Çizelge 2.3’de IOS 12 kullanıma sunulduktan sonra Kasım 2018’de IOS 11’in %27.1’e düştüğü ve IOS 12’nin ise %65.2 kullanım oranına ulaştığı görülmektedir [56]. Yeni sürümlerdeki kullanım oranlarının yüksek olmasındaki sebep Apple’ın yeni satışa çıkardığı cihazlarla birlikte yeni işletim sistemini piyasaya çıkararak, kullanıcıların önceki sürümleri yüklemesini önlemesi gösterilebilmektedir. Ayrıca önceki yıllarda satışa sunduğu cihazlara da aynı şekilde sürüm geri dönüşünü de kapatması bu oranların değişimini önemli ölçüde etkilemektedir.

Günümüzde dünya genelinde Apple'ın işletim sistemi IOS'u kullanan mobil cihazların kullanım oranları Şekil 2.1'de verilmiştir. Şekle baktığımızda son 3 yılda satışa sunulan iPhone cihazların, toplam kullanılan cihazların %43'ünü oluşturduğu görülmektedir.



Şekil 2.1. IOS kullanan cihazların dağılımı [56].

Çizelge 2.4'de verilen cihazlar, son sürüm işletim sistemini kullanan kullanıcı oranına göre sıralanmıştır. Çizelgede de görüldüğü gibi kullanıcıların büyük çoğunluğu son sürüm IOS'u kullanmayı tercih etmektedir. Son sıradakilerin yeni sürüm işletim sistemlerini hiç kullanmıyor olmalarının nedeni, cihazlarının bu sürümleri desteklemiyor olmasıdır.

Çizelge 2.4. iPhone kullanıcılarının IOS kullanım dağılımı [56].

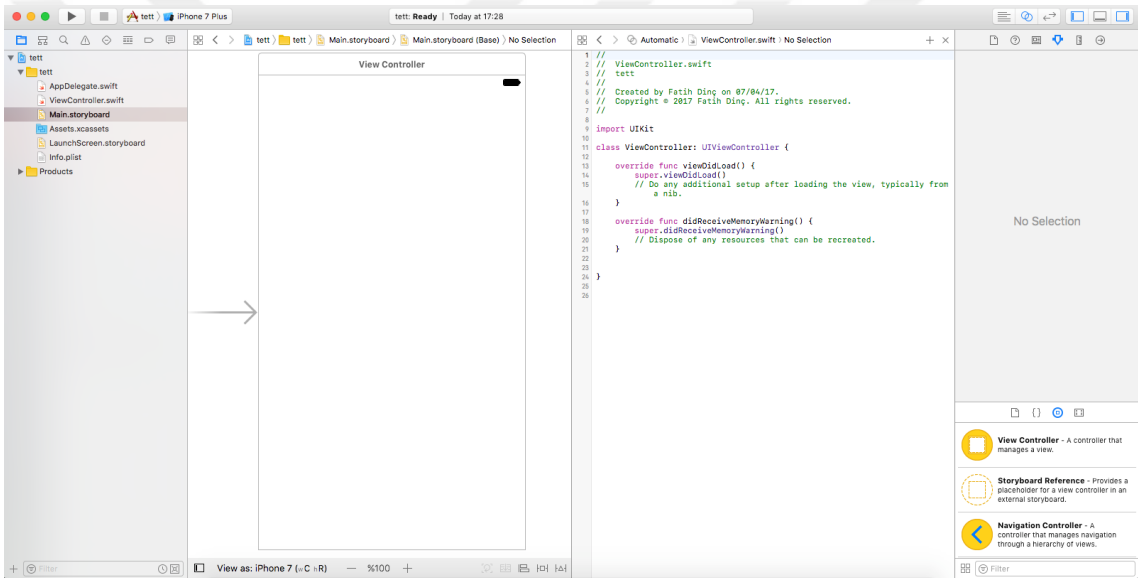
Cihaz	Cihaz Kullanımı	12.X	11.X	10.X	9.X	8.X	7.X	6.X
iPhone XS Max	%2.4	%100.0	%0.0	%0.0	%0.0	%0.0	%0.0	%0.0
iPhone XS	%1.0	%100.0	%0.0	%0.0	%0.0	%0.0	%0.0	%0.0
iPhone XR	%0.3	%100.0	%0.0	%0.0	%0.0	%0.0	%0.0	%0.0
iPhone X	%7.0	%93.0	%7.0	%0.0	%0.0	%0.0	%0.0	%0.0
iPhone 7+	%7.5	%83.4	%15.2	%1.4	%0.0	%0.0	%0.0	%0.0
iPhone 8	%5.7	%75.5	%24.5	%0.0	%0.0	%0.0	%0.0	%0.0
iPhone 7	%12.9	%75.0	%24.4	%0.6	%0.0	%0.0	%0.0	%0.0
iPhone 8+	%7.3	%70.3	%29.7	%0.0	%0.0	%0.0	%0.0	%0.0
iPhone 6s+	%4.1	%63.9	%35.6	%0.5	%0.0	%0.0	%0.0	%0.0
iPhone 6s	%18.1	%61.9	%33.2	%4.6	%0.2	%0.0	%0.0	%0.0
iPhone 5s	%5.9	%61.9	%24.8	%13.0	%0.2	%0.2	%0.0	%0.0
iPhone 6	%13.1	%60.8	%37.1	%1.7	%0.4	%0.0	%0.0	%0.0
iPhone se	%8.4	%54.9	%44.3	%0.8	%0.0	%0.0	%0.0	%0.0
iPhone 6+	%4.7	%42.6	%56.6	%0.7	%0.1	%0.0	%0.0	%0.0
iPhone 5c	%0.7	%0.0	%0.0	%97.6	%0.0	%2.4	%0.0	%0.0
iPhone 5	%0.5	%0.0	%0.0	%97.9	%0.0	%0.7	%0.0	%1.4
iPhone 4S	%0.3	%0.0	%0.0	%0.0	%72.9	%0.2	%26.9	%0.0
iPhone 4	%0.1	%0.0	%0.0	%0.0	%0.0	%0.0	%97.5	%2.5
iPhone 3GS	%0.1	%0.0	%0.0	%0.0	%0.0	%0.0	%0.0	%100.0
Toplam		%67.6	%28.5	%3.3	%0.3	%0.0	%0.2	%0.1

2.2. XCODE

Apple, entegre geliştirme ortamı (IDE) olarak sadece MacOS X işletim sisteminde çalışan Xcode adlı yazılım geliştirme platformunu tercih etmektedir. Herhangi bir ücret ödemeden kullanabilen Xcode ile iPhone ve iPad uygulamalarının yanı sıra Apple bilgisayarlarda çalışmak üzere MacOS X uyumlu programlar da geliştirilebilmektedir. Geliştirilen programlar simülatörler aracılığıyla test edilebilir olduğundan, gerçek bir cihaz üzerinde karşılaşılabilecek çeşitli durumları (bellek kaçakları, cihazın yatay

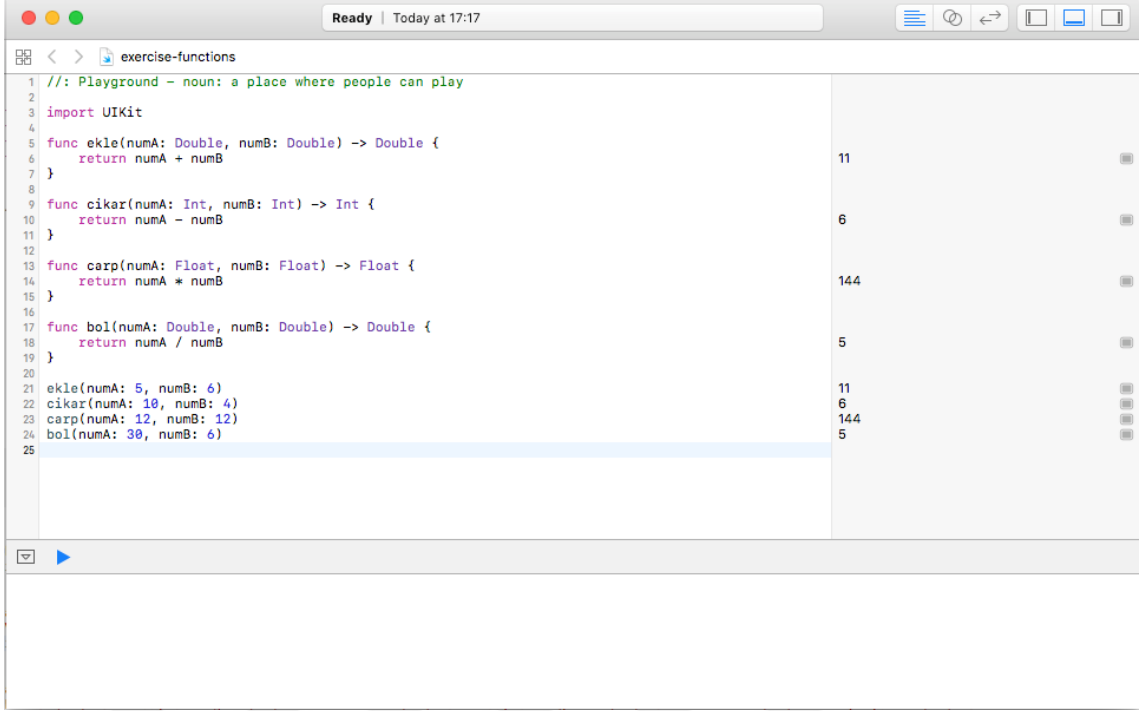
konumda tutulması vb.) test edebilmektedir. Uygulama geliştirme sürecinde geliştiricinin elinde farklı özelliklerdeki iPhone, iPad, iPod ve Mac cihazları bulunmadığı durumlarda, geliştirilen yazılım uygun simülatörde denenerak sorunlar tespit edilebilmektedir.

Şekil 2.2’de Xcode arayüzü görülmektedir. Maalesef Xcode’un Linux ve Windows işletim sistemleri için ayrı bir sürümü bulunmamaktadır. Bu yüzden Xcode ile geliştirme yapacak kişi, MacOS X yüklü bir bilgisayar sahibi olmak durumunda kalmaktadır. Sanal makina kullanarak Windows üzerinde MacOS X işletim sistemi kullanmak mümkündür. Bu yöntemle çalışan MacOS X sürümleri internet üzerinde kolaylıkla bulunabilmektedir. Ancak gerek donanım uyum sorunları gerekse de sanal makina üzerinde işletim sisteminin yavaş çalışması nedeniyle, bu yöntemin kullanılması geliştiriciye problem oluşturmaktadır.



Şekil 2.2. Xcode arayüzü.

IOS uygulama geliştirilirken, 2014’e kadar programlama dili olarak sadece Objective-C kullanılmaktaydı. 2014’de WWDC’de yeni yazılım dili Swift’in de kullanılacağı duyurulmuştur. Böylece geliştiriciler iki programlama dili kullanılabilir hale gelmiştir. Swift ile birlikte Xcode’a gelen en önemli yeniliklerden bir tanesi Playgrounds oldu. IOS’a uygulama geliştirenler, derleme konusunun aslında can sıkıcı bir durum olabildiğini çok iyi bilmektedirler. Geliştiriciler Swift ile birlikte gelen yeni Xcode sürümüyle birlikte artık gerçek zamanlı olarak projelerinin mevcut durumunu kod yazılan ekranda görebilmektedirler. Bununla birlikte ana kodu değiştirmeden, Playgrounds üzerinde yazılan kodu düzenleyerek yapılacak değişikliklerin nasıl tepki vereceğini Şekil 2.3’de gösterildiği gibi anlık olarak görmek mümkün olmaktadır.



Şekil 2.3. Örnek playground görüntüsü.

2.3. VERSİYON KONTROL SİSTEMİ VE KULLANIMININ FAYDALARI

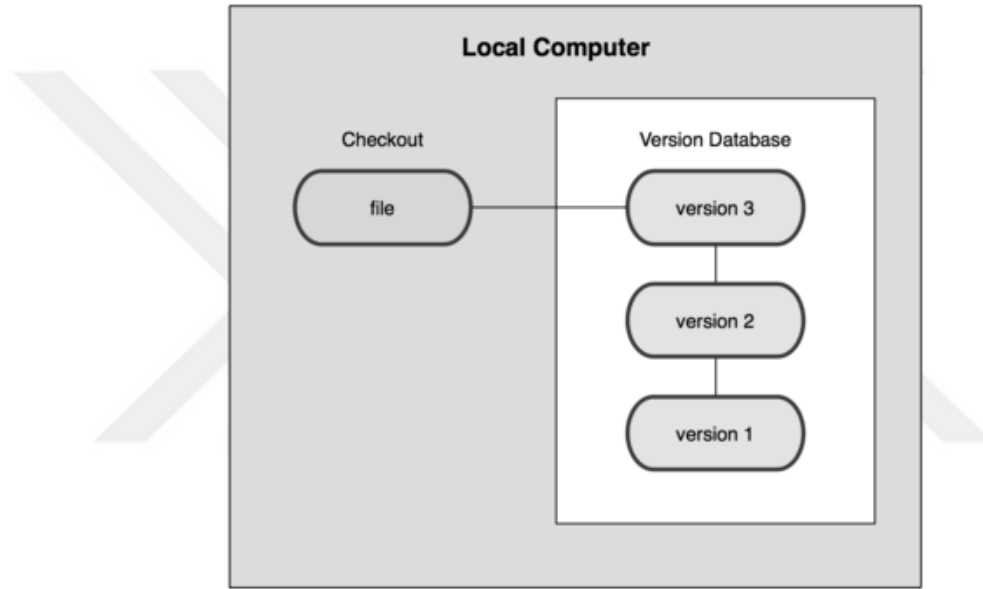
Çoğu kullanıcı, dosyaları klasöre kopyalayıp, klasörleri isimlendirirken tarih ile zaman bilgisini de ekleyerek sürüm kontrolü yapmayı tercih etmektedir. Bu tarz kullanım çok yaygın olarak uygulanmaktadır. Çok kolay bir şekilde kullanılmakla beraber, aynı zamanda da hatalara çok açık bir yöntem olabilmektedir. Hangi klasörde olduğu unutulup yanlış dosyaya yazılabilir veya dosyaların üstüne kopyalama yapılabilmektedir. Uzun zaman önce programcılar, bu sorunu ortadan kaldırmak için, dosyaların içindeki bütün değişiklikleri sürüm kontrolü altına alan, basit bir veritabanını kullanan versiyon kontrol sistemini (VKS) geliştirmişlerdir.

VKS, bir veya daha fazla dosya üzerinde yapılan değişiklikleri kaydeden ve daha sonra kaydedilen herhangi bir sürüme geri dönebilmeyi sağlayan sisteme verilen isimdir. VKS, dosyaların veya bütün projenin daha önce kaydedilmiş belirli bir sürümüne ulaşmanızı, zaman içinde yapılmış olan değişiklikleri karşılaştırmanızı, karşılaşılan bir soruna sebep olan şeyde en son hangi kullanıcının değişiklik yaptığını, karşılaşılan hatayı kimin, ne zaman sisteme dahil ettiğini ve daha birçok şeyin tespit edilebilmesini sağlamaktadır. Diğer yandan, bu sistemi kullanmak, bir hata yapıldığında veya bazı dosyalar yanlışlıkla silindiğinde durumun kolayca telâfi edilmesine imkân sağlamaktadır. Hatta, bütün bunlar kullanıcıya fark edilebilecek kadar ek yük getirmez.

Versiyon kontrolü; yerel versiyon kontrol sistemi, merkezi versiyon kontrol sistemi ve dağıtılmış versiyon kontrol sistemi üzerinden yapılabilir.

2.3.1. Yerel Versiyon Kontrol Sistemi

Yerel bilgisayarda proje için yapılan her commit mesajını bir versiyon olarak tutmakta ve bu commit değerine hash ataması yapılabilmektedir. Hash ataması sayesinde her versiyon bir diğerinden ayırt edilebilmektedir. Tarihsel sıra ile tutulan versiyonlar arası geçiş yapmak da bir git komutu kadar kolay olmaktadır. Ayrıca versiyon geçmişini görüntüleme imkânı sağlamaktadır. Şekil 2.4’de yerel versiyon kontrol sistemi diyagramı gösterilmektedir.



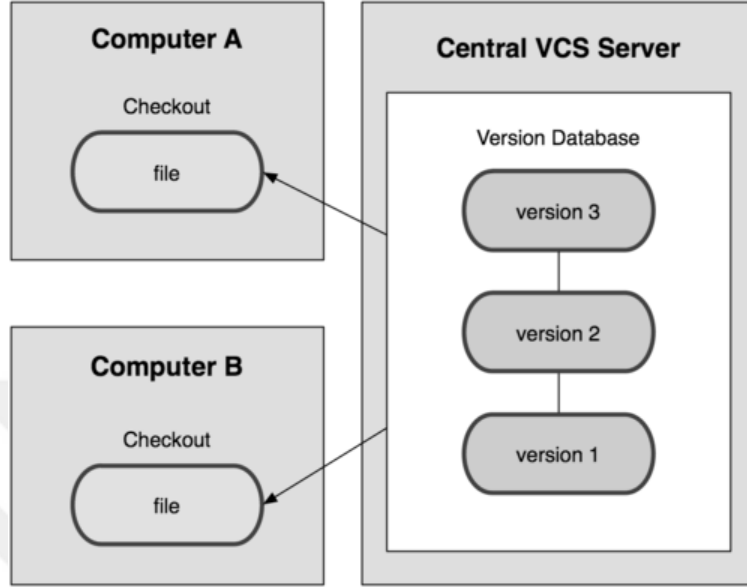
Şekil 2.4. Yerel versiyon kontrol sistemi diyagramı [57].

2.3.2. Merkezi Versiyon Kontrol Sistemi

Merkezi VKS yönteminin, özellikle yerel VKS'lere göre, birçok avantajı bulunmaktadır. Örneğin, çalışılan projedeki geliştiriciler, diğerlerinin yaptıklarından belirli ölçüde haberdar olmaktadır. Sistemi yöneten kullanıcı, projede çalışan geliştiricinin hangi yetkilere sahip olacağını oldukça detaylı bir biçimde düzenleyebilmektedir. Hatta merkezi VKS'leri yönetmek, her kullanıcının cihazında ayrı bir şekilde kurulmuş olan yerel veritabanlarını daha kolay bir şekilde yönetmeyi sağlamaktadır.

Şekil 2.5’de diyagramı gösterilen merkezi VKS’lerin de ciddi bazı problemleri bulunmaktadır. En büyük problem, merkezi sunucunun arızalanması halinde ortaya çıkabilecek kırılma noktasıdır. Merkezi veritabanının kayıtlı olduğu sabit diskin bozulması halinde, eğer yedeklemeler de dikkatli bir şekilde yapılmamışsa, projenin,

kullanıcıların bilgisayarlarında kalan yerel bellek kopyaları dışındaki bütün kopyalarının kaybedilme riski ile karşılaşmaktadır. Yerel VKS'ler de bu sorunla karşı karşıyadır. Projenin bütün kopyalarını, geliştiriciler tek bir yerde tuttuğu sürece her şeyin kaybolma riski bulunmaktadır [57].



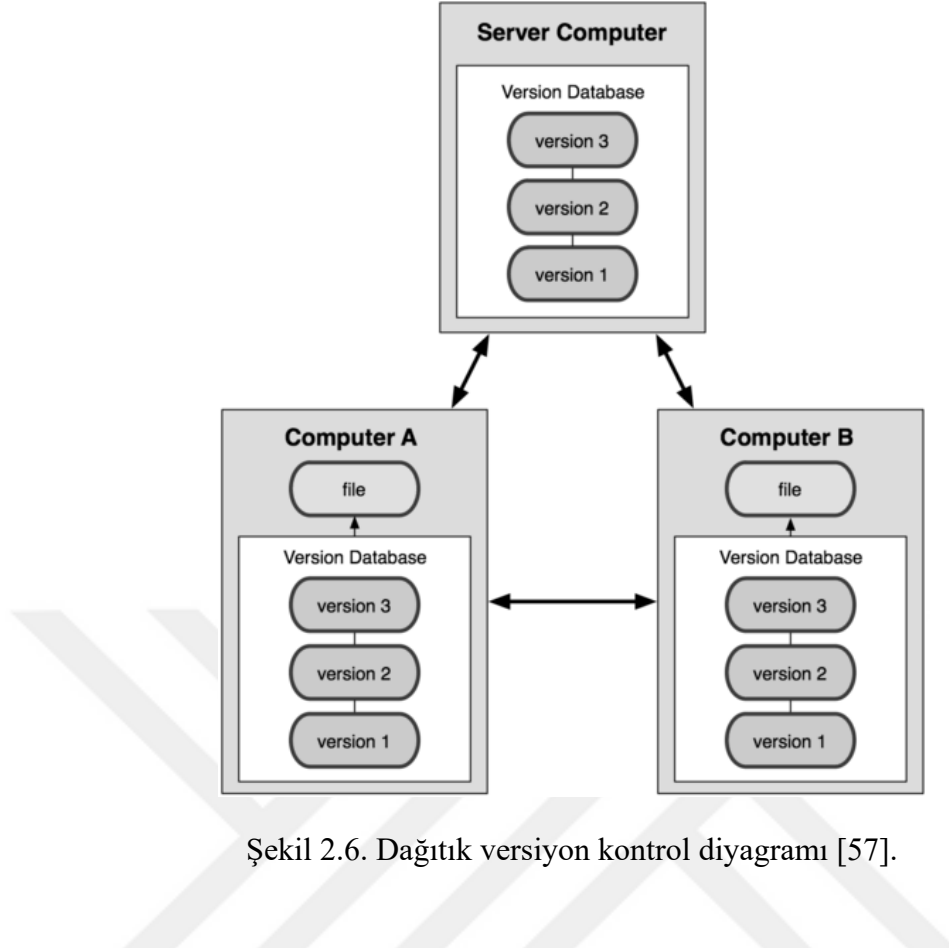
Şekil 2.5. Merkezi versiyon kontrol diyagramı [57].

2.3.3. Dağıtık Versiyon Kontrol Sistemi

Diğer iki yöntemdeki veri kaybı riskine karşı, dağıtık versiyon kontrol sistemleri devreye girmektedir. Bir dağıtık VKS'de (Mercurial, Git gibi), geliştirici çalıştığı proje dosyalarının sadece en son kayıt kopyalarını almakla kalmaz, yazılım havuzunu (repository) tamamen kopyalamaktadır.

Bu sayede, sunuculardan birinde problem olursa ve o sunucu üzerinde birlikte çalışma yürüten sistemler varsa, geliştiricilerden birinin yazılım havuzunu, çalışmaların yedeklendiği sunucuya geri yükleyerek sistemin kurtarılması sağlanmaktadır.

Şekil 2.6'da gösterilen dağıtık VKS sayesinde birbirinden çok uzakta bulunan birimlerdeki kullanıcılar yazılım havuzuyla rahat bir şekilde çalışabilmekte ve bu sayede aynı proje için farklı kişilerle ortak çalışmalar yürütebilmektedirler.



Şekil 2.6. Dağıtık versiyon kontrol diyagramı [57].

2.4. CORE DATA

Kalıcı verilerden bahsedildiğinde, genellikle akla ilk olarak veritabanı gelmektedir. Bu yüzden Core Data, veritabanıyla karıştırılmaktadır. SQLite veritabanı, iPhone’da Core Data için varsayılan kalıcı hafızadır. Core Data model katmanında çalışmaktadır. Aslında uygulama geliştiricilerin, veriyi veritabanında nesne yönelimli bir şekilde depolamasına veya geri yüklemesine olanak tanıyan bir yapı olarak görev yapmaktadır. Geliştirici, Core Data ile geliştirdiği uygulamalardaki nesnelere, herhangi bir SQL kodu bilmeden veritabanındaki tablo kayıtlarıyla kolayca eşitleyebilmektedir.

Core Data, geliştirilen uygulamalarda model katman nesnelere yönetmek için kullanılan bir yapıdır. Nesne yaşam döngüsü ve nesne grafiği ile ilişkili ortak görevler için otomatikleştirilmiş çözümler sunmaktadır [58].

Core Data’nın SQLite veritabanına göre daha karmaşık bir veri saklama yapısı bulunmaktadır. MacOS platformundan sonra mobil işletim sistemi platformuna da taşınarak iPhone uygulamalarında da kullanılabilir hale getirilmiştir. Core Data, tablolar arasında ilişki kurma, karmaşık veritabanı sorguları yazma, bellek üzerinde saklama gibi özellikler sunarak uygulamalarda gelişmiş bir veritabanı sistemine ihtiyaç duyulduğu

durumlarda birçok ihtiyacı karşılamaktadır. Yapı olarak içerisinde SQLite kullanan Core Data, kütüphane içerisinde bulunan gömülü fonksiyonlar ile veritabanı işlemlerini gerçekleştirmektedir [59].

Core Data, model katmanını desteklemek için yazılacak olan kodun yüzde 50 ile 70 oranları arasında düşürülmesini sağlamaktadır. Bu özellik; uygulamak, test etmek veya optimize etmek zorunda olmadığımız aşağıdaki dahili özelliklere bağlı olmaktadır [58].

- Özellik değerlerinin otomatik olarak doğrulanması. Yönetilen nesnelere, değerlerin kabul edilebilir aralıklarda bulunmasını sağlamak ve değerlerin kombinasyonlarının mantıklı olmasını sağlamak için standart anahtar ve değer kodlama doğrulama yöntemlerini genişletir.
- Nesnelere ağır yüklenerek, yazma sırasında kopyalama yapılarak yük azaltılması,
- Kullanıcı arayüzü senkronizasyonunu sağlamak için kontrol katmanı ile isteğe bağlı entegrasyon yapılması,
- Şema değişikliklerini basitleştiren ve etkili bir şekilde geçiş yapılmasını sağlayan şema taşıma araçları,
- Bellekte ve kullanıcı arayüzünde veri gruplama, filtreleme ve düzenleme yapılması,
- Gelişmiş sorgu derleme yapılması. SQL sorgu yazmak yerine NSPredicate nesnesinin veri getirme isteğiyle (fetch request) karmaşık sorgular oluşturulabilmesi,
- Versiyon izleme ve kilitleme kullanılarak otomatik olarak birden fazla geliştiricinin çatışma çözümlemesinin yapılması,

özellikleri ile kod azaltılması sağlanmaktadır [58].

Core Data, uygulama içerisinde kullanılan verilerin, mobil cihaz hafızasında oluşturulan veritabanına kaydedilmesini ve kullanılmasını sağlamaktadır. Verileri sürekli olarak sunucudan çekerek kullanıcıya sunmak performans açısından iyi bir yöntem olmamaktadır. Eski verileri Core Data hafızasına alıp, yeni veri geldikçe bunlar sunucudan çekilmelidir. Böylelikle uygulamanın hızlı, internet ihtiyacı olmadan ve efektif bir şekilde kullanılabilmesi sağlanabilmektedir. Fakat bu işlem, uygulama boyutunun artmasına sebep olmaktadır.

Mobil cihazlarda, bazı uygulamaların sürekli olarak boyutunun artmasının sebebi,

içlerinde bulunan yerel veritabanına sürekli olarak yeni verilerin eklenmesidir.

Güncelleme yapılacağı zaman, veritabanı birleştirme işlemi doğru yapılmazsa, güncelleme sonrası kullanıcı uygulama çökmesi ile karşılaşabilmektedir. Bu problemi aşmak için kullanıcılar, uygulamayı yeniden yüklemek zorunda kalmaktadırlar. Bu nedenle kesinlikle dikkat edilmesi gereken bir durum haline gelmiştir.

Çizelge 2.5’de veritabanı ile Core Data’nın yaptıkları işler ve nasıl yaptıklarıyla ilgili bilgiler verilmiştir.

Çizelge 2.5. Veritabanı ile Core Data'nın karşılaştırılması [60].

Veritabanı	Core Data
Birincil işlevi veri saklamak ve almaktır	Birincil görevi grafiksel yönetimidir. (Diske okuma ve yazma destekleyici özelliştir.)
Diskte saklanan bilgi ile çalışır (Minimum veya artırılmış olarak yüklenir)	Bellekte saklanan nesnelere üzerinden çalışır.
‘dump’ verilerini saklar	İşlevlerini kendi yöneten, alt sınıflara ayırabilen ve sonraki görevler için özelleştirebilen nesnelere çalışır.
İş parçacıklı ve çok kullanıcıya olabilir	Tek iş parçacıklı ve tek kullanıcıdır (Bunları sağlayan Core Data etrafında soyutlama oluşturulmadığı sürece)
Tabloları silebilir ve veriyi belleğe yüklemeyi düzenleyebilir	Sadece bellekte çalışır
Sürekli diske kaydedilir	Kaydetme işlemi gerektirir
Milyonlarca yeni satır oluşturmak yavaş olabilir	Çok hızlı bir şekilde milyonlarca yeni nesne oluşturabilir (Yine de bu nesnelere kayıt etmek yavaş olabilir)
‘Unique’ anahtarlar gibi veri kısıtlamaları vardır	Veri kısıtlamalarını programın iş mantığına bırakır

2.5. SKETCH İLE TASARIM

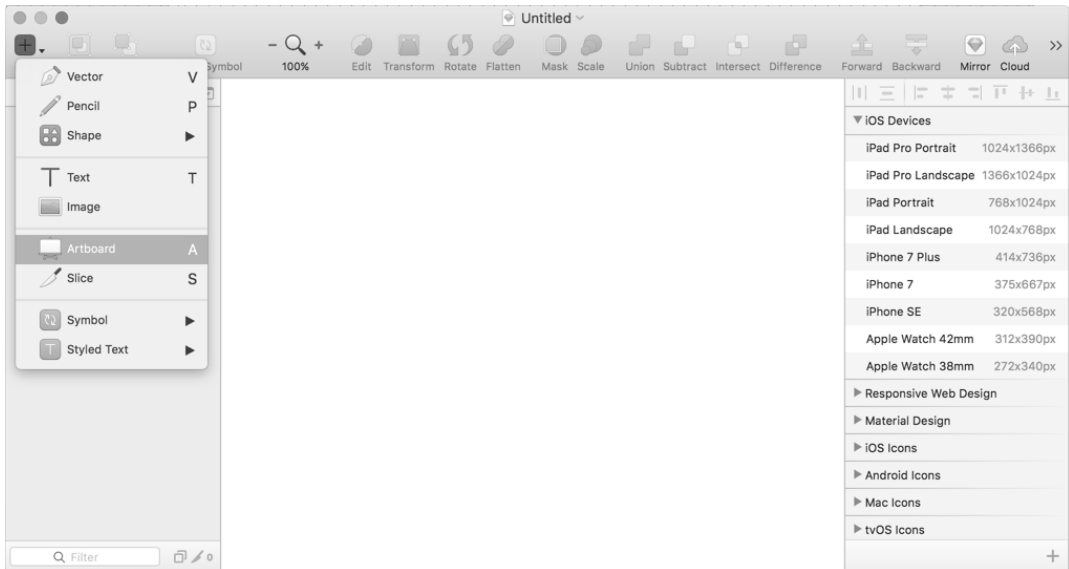
İlk kez 2010’da Hollandalı şirket Bohemian tarafından piyasaya sürülen Sketch, 2012’de Apple tasarım ödülünü kazanmıştır. Ülkemizde de yavaş yavaş yayılmaya başlamaktadır. Bu sürecin yavaş işleminde, sadece Apple’ın işletim sistemi MacOS’da çalışıyor olmasının büyük etkisi olmaktadır. Bunun da Sketch’in kullanımının yaygınlaşması üzerinde olumsuz bir etkisi olmaktadır. Türkiye gibi ülkelerde, hâlâ birçok şirketin Apple bilgisayarlarını satın alacak bütçeye sahip olmamaları, kullanımının çabuk yayılmasında

bir engel olarak ortaya çıkmaktadır.

Sketch'in kullanılmasına, özellikle Adobe Photoshop'un dijital tasarım işlerinde hantal kalması sebep olarak gösterilebilmektedir. Sketch, hızlı, dinamik ve daha basit bir uygulamadır. Diğer artı özelliği, arayüz iskeleti (wireframe) ya da taslak (mock-up) çizmek için ek bir tasarım programına ihtiyaç duyulmamasıdır. Çünkü en az o taslak programları kadar basit ve hızlı çalışmaktadır. Ayrıca arayüz iskeletlerini yüksek detaylı (high-fidelity) olarak çizmek gerekirse Sketch bu konuda diğer arayüz iskeleti uygulamalarından daha hızlı ve detaylı çalışabilmektedir.

Web, mobil, kullanıcı deneyimi/kullanıcı arayüzü (UX/UI) işleminin yanında gerektiğinde simge tasarlamak için de kullanılabilir. Bir diğer önemli özelliği ise birçok eklenti (plug-in) desteğinin olması ve onların da tasarım yaparken işinizi hızlandırmasına yardımcı olması gösterilebilmektedir.

Şekil 2.7'de, Sketch programında çalışma yüzeyi eklemek istediğimizde sağ tarafta gelen menüde iPhone, iPad, Mac, Android ve web için yapacağımız tasarımlar için kullanabileceğimiz kayıtlı ölçülerde çalışma yüzeyleri görülmektedir. Ayrıca hazır şablonlar arasında simge (icon) boyutları da bulunmaktadır. Uygulamadaki hazır şablonlar sayesinde tasarımını yapacağımız cihazın ölçülerinin bilinmesine ve ölçülendirilmesine gerek kalmayıp, tasarım aşamasında geliştiricinin işi kolaylaştırılmaktadır.



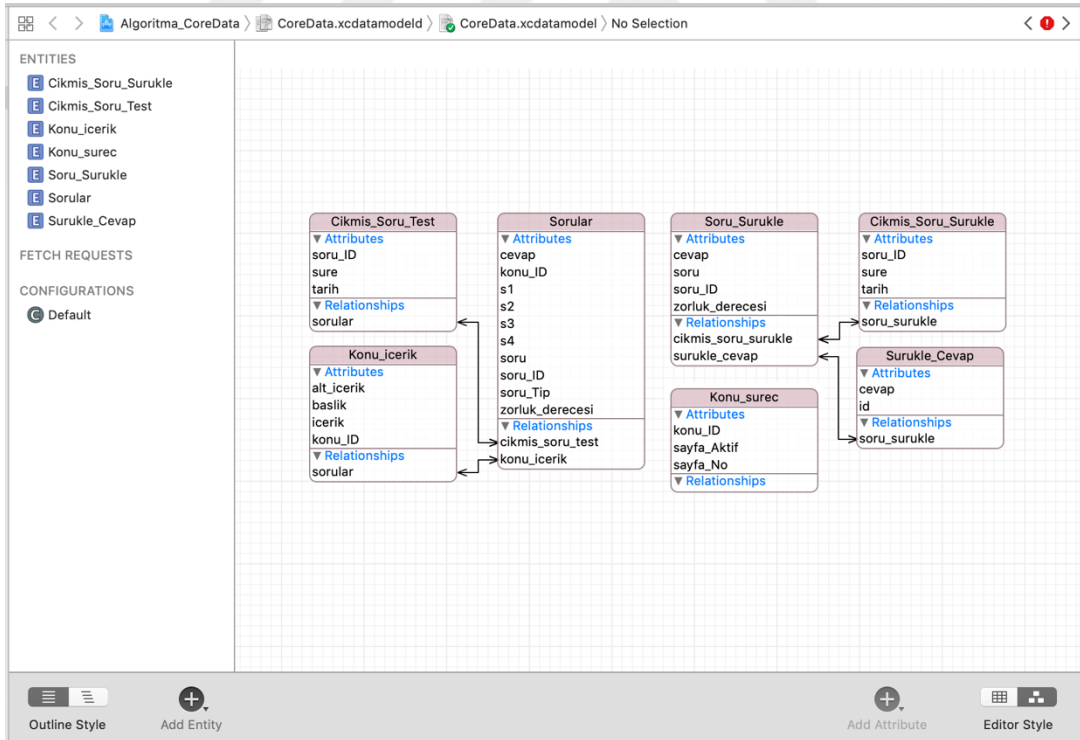
Şekil 2.7. Sketch çalışma yüzeyi ekleme.

2.6. ALGORİTMA EĞİTİMİ İÇİN UYGULAMANIN GELİŞTİRİLMESİ

Bu tez çalışmasında, algoritma eğitimine yardımcı olmak amacıyla “Algoritma Öğren” adında IOS uygulaması geliştirilmiştir. Uygulama geliştirilirken, program kodları yazımında Xcode uygulaması ve Swift programlama dili, tasarımında Sketch programları kullanılmıştır. Core Data veri modeli ile veri kaydı yapılarak kullanıcıların derslerdeki süreçleri kayıt altına alınmaktadır. Sürüm kontrolü için yerel VKS kullanımı tercih edilmiştir.

2.6.1. Veritabanı

Uygulamanın bilgileri veritabanına Core Data aracılığıyla kayıt edilmektedir. Uygulamaya Core Data eklendikten sonra Şekil 2.8’de görülen Xcode ekranının alt kısmından Add Entity menüsü ile tablolar oluşturulmuş, Add Attribute menüsü ile de tablolara alanlar eklenmiştir. Geliştirdiğimiz uygulamada, 7 adet tablo bulunmaktadır. Bu tablolar Core Data ile oluşturulduğunda otomatik olarak SQLite tabloları da oluşturulmaktadır.



Şekil 2.8. Core data tabloları.

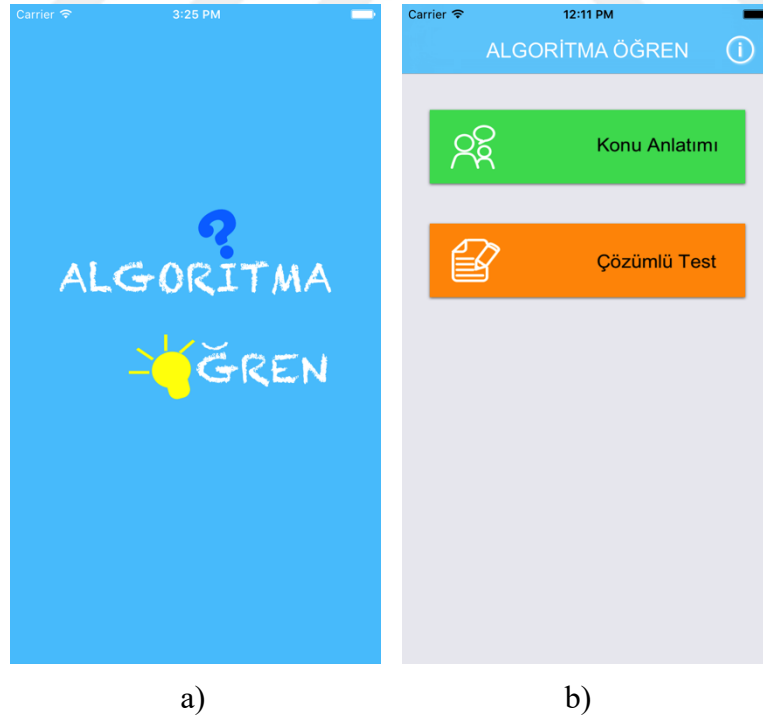
Kullanılan tabloların hangi bilgilerin kaydedilmesini sağladığı Çizelge 2.6’da gösterilmektedir.

Çizelge 2.6. Tablolar ile tutulan bilgiler.

Tablo Adı	Bilgiler
Konu_icerik	Konu anlatımında kullanılan bilgiler
Konu_surec	Kullanıcının konu çalışırken geldiği son konum ve aktif bulunduğu konum bilgileri
Sorular	Kullanıcıya sorulan, test tipi soruların bilgileri
Cikmis_Soru_Test	Kullanıcının karşılaştığı test soruları ve cevaplama süresi
Soru_surukle	Kullanıcıya sorulacak akış diyagramı soruları
Cikmis_Soru_Surukle	Kullanıcının karşılaştığı diyagram soruları ve cevaplama süresi
Surukle_cevap	Kullanıcıya sorulan akış diyagramı sorularının cevapları

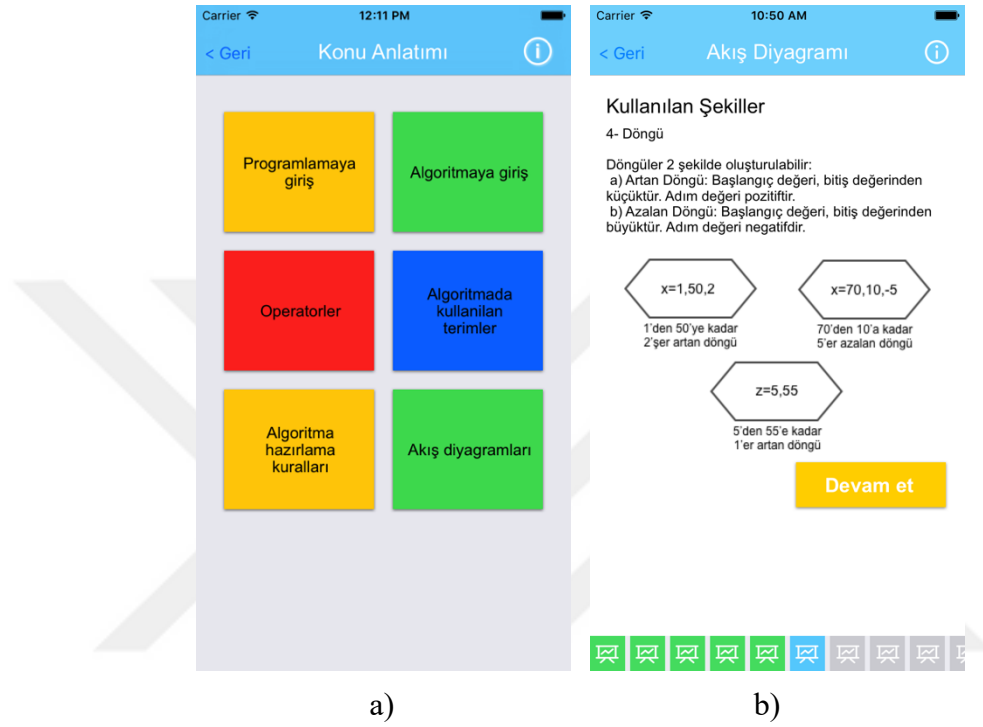
2.6.2. Uygulamanın Kullanımı

Şekil 2.9’da “Algoritma Öğren” uygulamasının başlama ve ana menü ekran görüntüsü bulunmaktadır. Başlama ekranı, uygulama açılırken kısa bir süre görüntülendikten sonra kaybolmakta ve kullanıcıyı ana menü karşılamaktadır.



Şekil 2.9. Uygulama ekranları a) Başlama ekranı b) Ana menü.

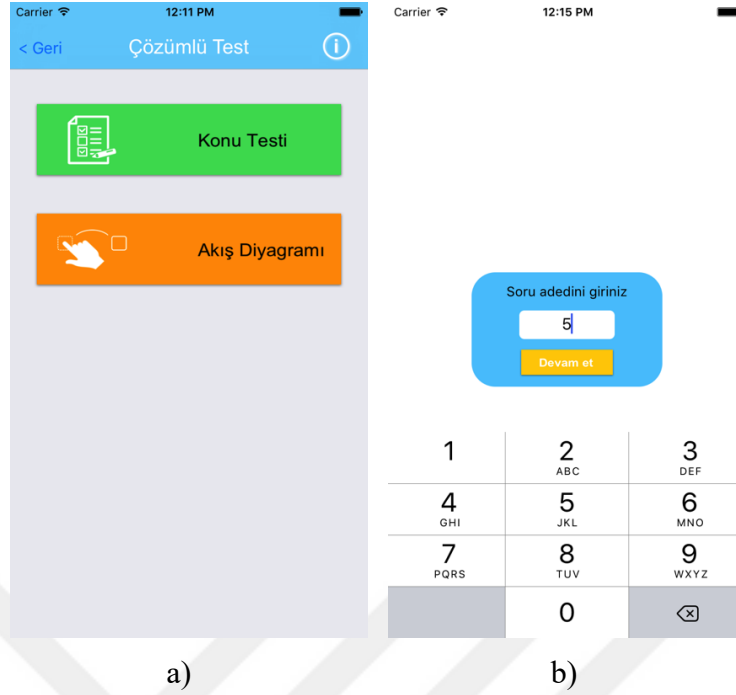
Şekil 2.9.a’da başlama ekranı (launch/splash screen) ve Şekil 2.9.b’de ise uygulamanın kullanıcıyı karşılayan ana menüsü bulunmaktadır. Kullanıcı menü ekranında bulunana bu iki seçenek sayesinde konu anlatımı ve sorulara hızlı bir şekilde geçiş yapabilmektedir. Konu anlatımı menüsüne girildiğinde kullanıcıyı Şekil 2.10.a’daki konu başlıklarının bulunduğu menü karşılamaktadır. Buradan kullanıcı çalışmak istediği konuyu seçerek çalışmasına başlayabilmektedir.



Şekil 2.10. Konu anlatımı a) Menü b) İçerik.

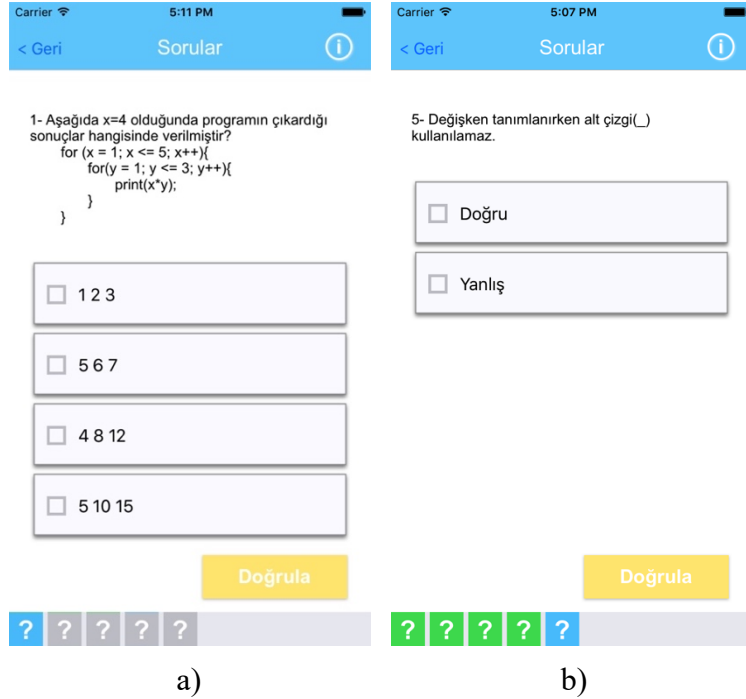
Şekil 2.10.b’de “Akış diyagramları” konusuna giren kullanıcının karşılaşacağı içeriklerden örnek ekran gösterilmektedir. Kullanıcının daha önce hiç girmediği konular gri simgeler ile gösterilmekte ve bu konulara “Devam et” butonuyla sıralı olarak girebilmektedir. Çalıştığı konular yeşil simgelerle gösterilmekte ve bu konular arasında simgeye tıklayarak istediği konuyu tekrar açabilmektedir. Aktif olarak çalıştığı konu ise mavi simge ile gösterilmektedir. Böylece kullanıcı nerede olduğunu anlayabilmektedir.

Şekil 2.9’da bulunan “Çözümlü Test” menüsünü seçildiğinde kullanıcıyı Şekil 2.11.a’daki gibi soru tiplerinin bulunduğu menüler karşılamaktadır. “Konu Testi” seçeneğinden kullanıcı teorik bilgilerini test edebilirken, “Akış Diyagramı” menüsünden soruların karşılığı olan akış diyagramlarını oluşturarak pratik yapabilmektedir. Şekil 2.11.b’deki şekilde ise test menülerine girildiğinde karşımıza çıkan seçenek görülmektedir. Buradan kullanıcıya kaç soru istediği sorulmakta ve istenilen adette daha önceden girilmiş olan sorular arasından rastgele seçilerek getirilmektedir.



Şekil 2.11. Çözümlü testler a) Test menüsü b) Soru adedi.

Şekil 2.12.a ve 2.12.b'de kullanıcıya konu testi soru bankasından sorulan sorulardan örnek ekran görüntüleri verilmiştir. Kullanıcı konu testine girdiğinde sorular rastgele seçilerek kullanıcıya sorulmaktadır. Böylece kullanıcının sürekli aynı sorularla karşılaşma ihtimali düşürülmektedir.



Şekil 2.12. Konu testleri a) Çoktan seçmeli b) Doğru/Yanlış.

Kullanıcının cevaplamadığı sorular gri simgeler ile gösterilmektedir. Sorulara ister “Doğrula” butonuyla sıralı olarak isterse simgesine tıklayarak girebilmektedir. Cevapladığı sorular yeşil simgelerle gösterilmektedir. Aktif olarak cevapladığı soru ise mavi simge ile gösterilmektedir. Böylece kullanıcı hangi soruda olduğunu anlayabilmektedir.

Şekil 2.13’de kullanıcının sorulan sorulara verdiği cevaplar ve soruların doğru cevapları gösterilmektedir. Yanlış verilen cevaplar sarı arka fon rengi ve kırmızı yazı rengi ile renklendirilerek gösterilmektedir. Bu sayede kullanıcı yanlış verdiği cevapları hemen tespit edip, doğru cevabı öğrenebilmektedir.



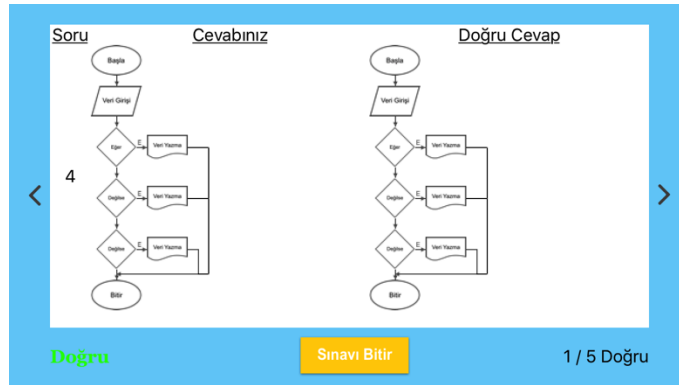
Şekil 2.13. Konu testi cevap anahtarı.

Kullanıcı, Şekil 2.11.a’da verilen çözümlü test ekranından “Akış Diyagramı” menüsüne girildiğinde ekrana gelen soru sayısı ekranını geçtikten sonra kullanıcı Şekil 2.14’de verilen akış diyagramı sorularıyla karşılaşmaktadır. Buradan kullanıcı sağ tarafta verilen akış diyagramında kullanılan şekilleri kullanarak sorunun cevabını oluşturmaktadır. Cevaplar sürükle-bırak yöntemiyle oluşturulabilmektedir.



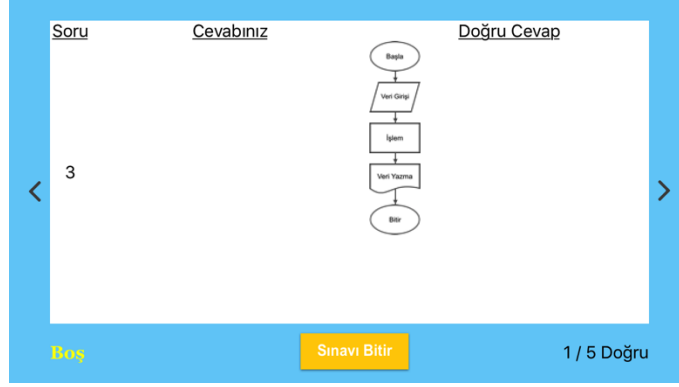
Şekil 2.14. Akış diyagramı soru ekranı.

Kullanıcı akış diyagramı sorularını cevaplandırdıktan sonra Şekil 2.15’de gösterildiği gibi, kullanıcının verdiği cevaplar ve doğru cevaplar kendisine gösterilerek geri bildirim yapılmaktadır. Ekranın sağ alt köşesinde kullanıcının verdiği doğru cevap/toplam soru sayısı gösterilirken, sol alt köşesinde ise soruya verilen cevabın doğru olduğu Şekil 2.15.a’da, cevap verilmeden geçildiği 2.15.b’de ve yanlış cevap verdiği 2.15.c’de gösterilmektedir.

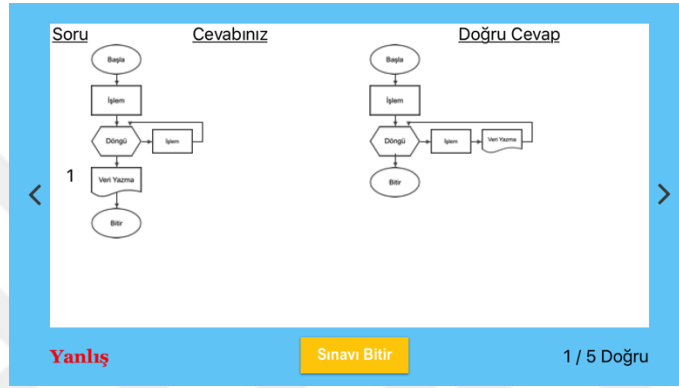


a)

Şekil 2.15. Akış diyagramı cevap anahtarı a) Doğru b) Boş c) Yanlış.



b)



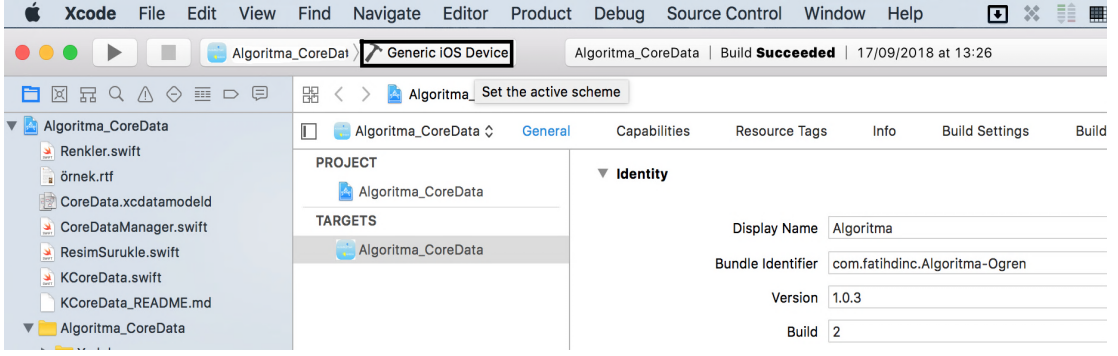
c)

Şekil 2.15. (devam). Akış diyagramı cevap anahtarı a) Doğru b) Boş c) Yanlış.

2.7. KULLANICILARIN UYGULAMAYI TEST EDEBİLMESİ İÇİN TESTFLIGHT KULLANILMASI

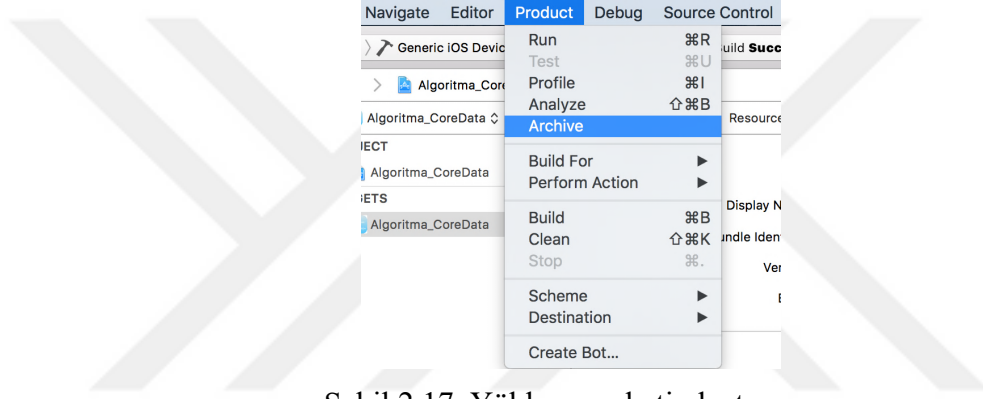
Geliştirilen uygulama, direkt bilgisayardan mobil cihazlara yüklenebileceği gibi internet üzerinden de yüklenebilir. Fakat kullanıcılara uygulamayı internet üzerinden test ettirebilmek için TestFlight uygulamasının kullanılması gerekmektedir. Bu hizmetin kullanılabilmesi için de geliştirici hesabının satın alınması gerekmektedir.

Geliştirici hesabı alındıktan sonra Xcode programından geliştirilen proje açılır ve sonra Şekil 2.16'da işaretli olan simülatör seçeneklerinden 'Generic IOS Device' seçeneği seçilir.



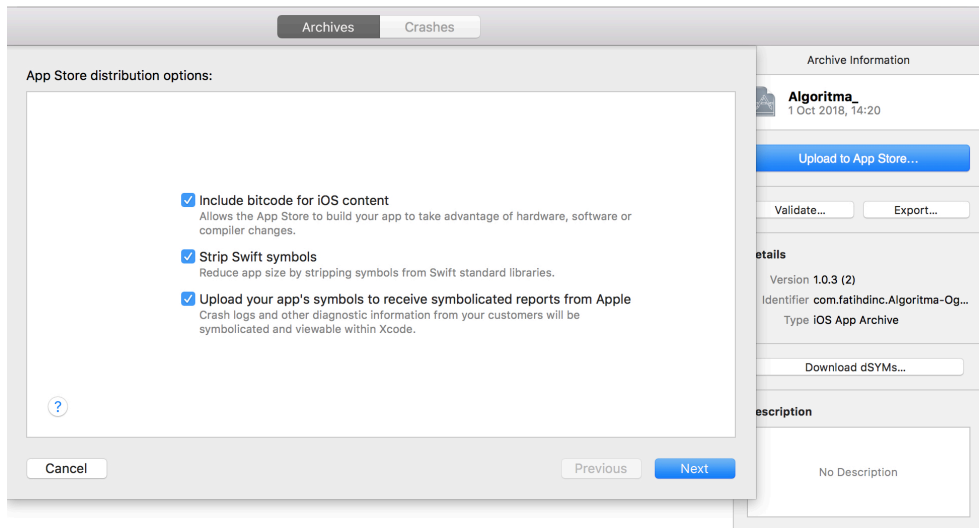
Şekil 2.16. Simülâtör seçimi.

Daha sonra Şekil 2.17’de gösterilen Archive menüsüne girilir. Buradan uygulama bir cihaza yükleniyormuş gibi paket oluşturulur ve internete yüklemeye hazır hale getirilir.



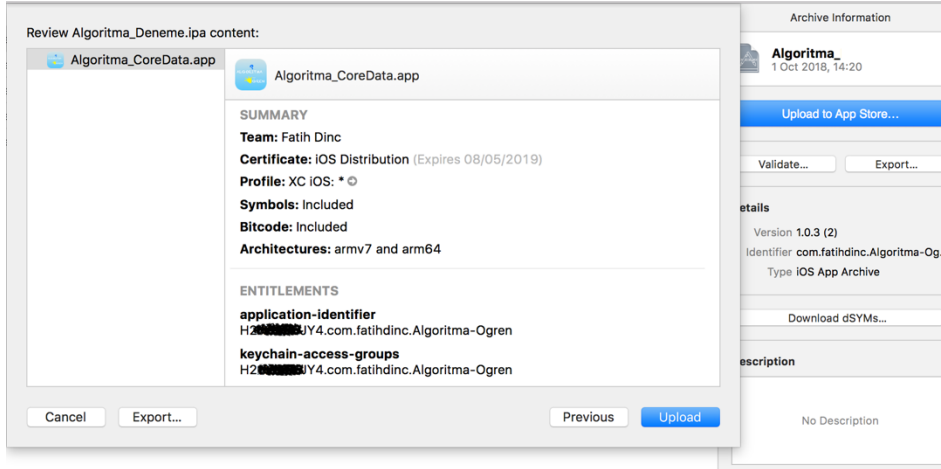
Şekil 2.17. Yükleme paketi oluşturma.

Uygulamanın derlenmesinde herhangi bir sorun yoksa Şekil 2.18’deki ekranla karşılaşılır. Karşılaşılan ekranlarda Next seçeneğiyle kurulum ilerletildiğinde son olarak Şekil 2.19’deki ekranla karşılaşılır.



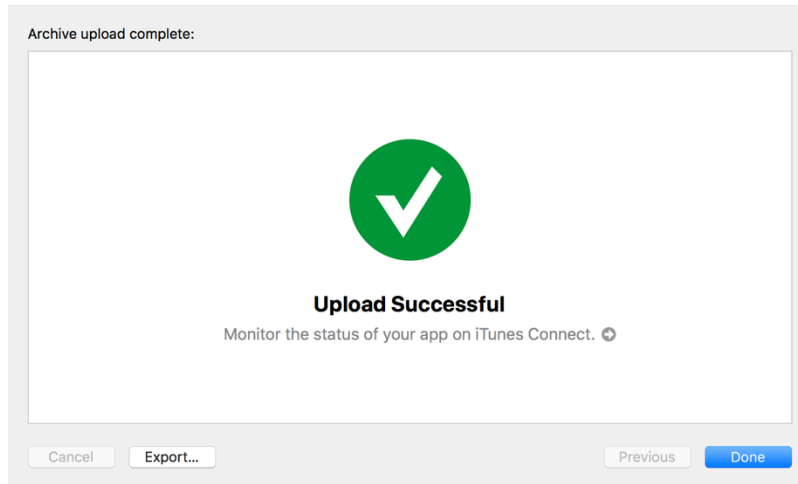
Şekil 2.18. Yükleme seçenekleri.

Şekil 2.19'deki ekrandan Upload'a tıkladığımızda uygulamanın Apple sunucularına yükleme işlemi başlatılır.



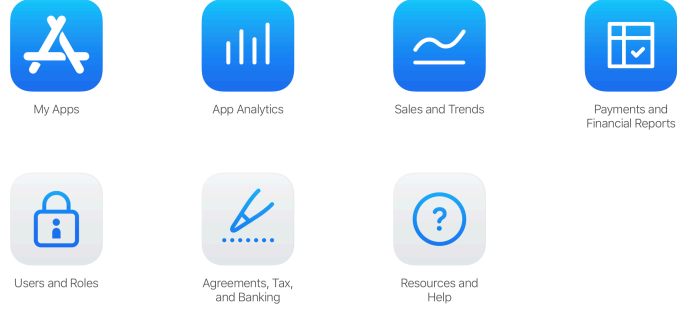
Şekil 2.19. Yükleme öncesi bilgilendirme.

İstenilen temel gereksinimlerin karşılanması durumunda uygulamaların yüklenme işleminde bir sorunla karşılaşılmayacak ve Şekil 2.20'deki ekranla karşılaşılacaktır. Yükleme işlemi tamamlandıktan sonra Apple tarafından incelenme süreci başlatılır ve onaylanıp onaylanmadığı geliştiriciye mail ile bildirilir. Onaylanması durumunda uygulama Appstore'a yüklenebilir veya test edecek kullanıcılara uygulamayı kurma yetkisi verme işlemine geçilebilir.



Şekil 2.20. Yükleme başarılı.

Onay süreci tamamlandıktan sonra geliştirici hesabımıza giriş yapıldığında Şekil 2.21'deki ekranla karşılaşılır ve "My Apps" menüsüne giriş yapılır. "My Apps" menüsüne girdikten sonra Şekil 2.21'deki gibi uygulama ile karşılaşılır.

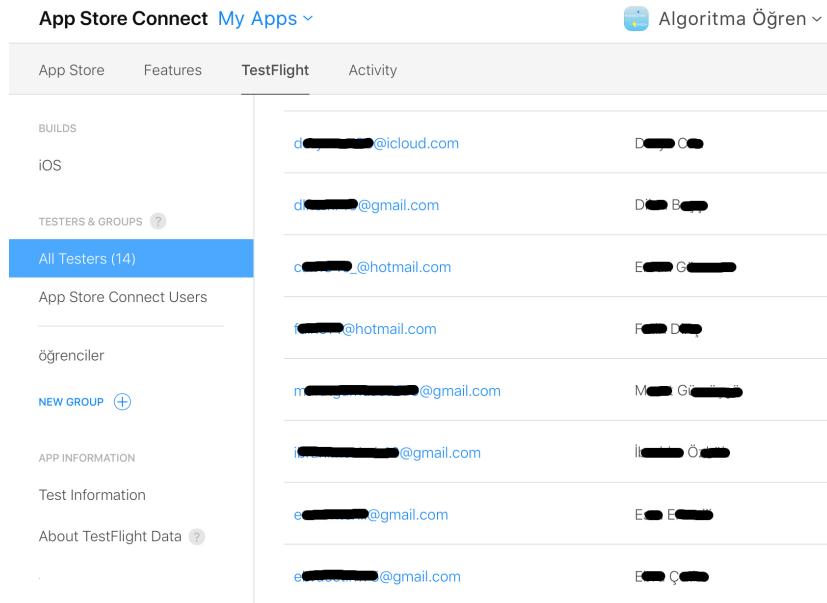


Şekil 2.21. iTunesConnect hesabı ana menü.



Şekil 2.22. Yüklenen uygulama.

Buradan da uygulamaya tıkladıktan sonra, Şekil 2.23’de görüldüğü gibi menülerden TestFlight sekmesine girildiğinde buradan uygulamayı test edecek olan kullanıcılar eklenerek maillerine yükleme linkleri gönderilebilir. Kullanıcılara gönderilen mailler, uygulamanın nasıl yüklenileceği ile ilgili bilgilendirmeleri de içerir. Kullanıcılar bu aşamadan sonra uygulamayı test edebilir duruma gelir ve bilgilendirmeleri takip ederek geri dönüş yapabilirler.



Şekil 2.23. TestFlight menüsü.

2.8. UYGULAMA AŞAMASI

Araştırmanın çalışma grubunu, Muş Alparslan Üniversitesi Bilgisayar Mühendisliği'nde öğrenim gören 40 öğrenci oluşturmaktadır. Bu bölümdeki öğrencileri seçmedeki amaç, algoritma eğitiminin bilgisayar mühendisliği için temel ve çok önemli bir konu olmasıdır. Çalışmanın deney grubunu 12, kontrol grubunu ise 28 kişi oluşturmaktadır. Kontrol grubunun deney grubundan fazla olmasının nedeni, öğrencilerin kullandıkları akıllı telefonların kullandığı işletim sisteminin IOS olmamasından kaynaklanmaktadır. Deney grubundaki öğrencilere çalışma sırasında uygulamanın nasıl kullanılacağı hakkında kısaca ön bilgi verilmiş, daha sonra öğrenciler başka uygulamalardan kazandıkları tecrübelerle uygulamayı kullanabilmişlerdir. Çalışma temel algoritma geliştirme konularıyla sınırlandırılmıştır. Araştırmaya katılan öğrenciler daha önce algoritma dersi eğitimini almışlardır.

Uygulama süreci, 2017/2018 eğitim-öğretim yılının bahar döneminde gerçekleştirilmiştir. Deney grubunda bulunan öğrenciler, kullandıkları akıllı telefonlardan algoritma öğren uygulamasını kullanarak eğitim almışlardır. Aynı zamanda da araştırmacı tarafından eğitim verilmiştir. Mobil öğrenme uygulaması henüz AppStore'a yüklenmediği için öğrencilerin yanında bilgisayardan direkt akıllı telefonlarına yüklemek yerine, Apple firmasının uygulama denenmesinde yardımcı uygulaması olan TestFlight üzerinden uygulamanın linki gönderilerek öğrencilerin geliştirilen uygulamayı kullanması sağlanmıştır. Bu yöntemin seçilme sebebi, uygulamanın Apple tarafından incelendikten sonra öğrencilerin yüklemesi sağlanarak, onların daha güvenli bir şekilde uygulamayı kullanmasını sağlamaktır. Yine bu uygulama aracılığıyla veya elektronik posta ile öğrencilerden geri bildirim alınmıştır.

Kontrol grubundaki öğrenciler, sadece araştırmacıdan eğitim almışlardır. İki gruptaki öğrencilere de başarı testi uygulanmıştır. Sorular doğru/yanlış, çoktan seçmeli ve klasik olarak hazırlanmıştır. Başarı testi, temel algoritma konularının içeriğine göre hazırlanmıştır. Bu öğrencilerden alınan yanıtlar sonrasında güvenilir sonuçlar ortaya koymak için denek sayısı 30'a düşürülmüştür. Çizelge 2.7'de çalışmaya katılan denek sayısı 30'a düşürüldükten sonraki başarı durumu gösterilmiştir.

Çizelge 2.7. Çalışma gruplarının başarı durumu

	Öğrenci Sayısı	Soru Sayısı		Not Ortalamaları		
		Test	Klasik	Test	Klasik	Başarı Notu
Kontrol Grubu	22	6	4	20	30.91	51.91
Deney Grubu	8	6	4	25	43.71	68.71

Başarı belirleme sınavı; test soruları 30 puan, klasik sorular 70 puan olacak şekilde hazırlanmıştır. Test soruları 6 adet 5 puanlık sorulardan oluşmaktadır. Klasik sorular ise 3 adet 15 puanlık ve 1 adet 25 puanlık soru olmak üzere toplam 4 sorudan oluşmaktadır. Çizelge 2.7’de çalışmaya katılan öğrenci gruplarının bu sınavdan aldıkları notlar; test türündeki not ortalamaları, klasik türdeki not ortalamaları ve toplamda aldıkların not ortalamaları gruplandırılarak gösterilmiştir.

3. SONUÇLAR VE ÖNERİLER

3.1. SONUÇLAR

Dijital yerli öğrenenler olarak adlandırılan genç nesil, teknoloji alanında ortaya çıkan yeniliklere hızlı bir şekilde uyum sağlamak ve bu gelişimleri hızlı bir şekilde kabullenmektedirler. Fakat, yeni çıkan teknolojilerin eğitimsel açıdan uygun olup olmamasına dikkat etmemektedirler. M-öğrenme ile ilgili olumlu görüşlere sahip olsalar da aslında mobil cihazlarını öncelikle iletişim ve sosyal ağlara bağlanma aracı olarak kullanmaktadır. Ödev yapma, not alma, kütüphane verilerine erişim vb. öğrenme amaçlı olan etkinliklerin kullanım oranı düşük kalmıştır. Mobil teknolojilerin gelişmesiyle ortaya çıkan imkanların anlık haberleşme ve sosyal ağlarda kullanılması, öğrenme ve öğretme uygulamalarının önüne geçtiği görülmüştür.

Teknolojideki gelişmelerin, eğitim alanında zenginleştirilmiş öğrenme ortamları sunduğu da görülmektedir. Bireyin duyu organlarına hitap edilerek gerçekleşen öğretimin kalıcı öğrenmelerin gerçekleşmesine etkisi büyüktür. Gelişen teknolojiler sayesinde bireyler öğrenme sürecini mekâna ve zamana bağlı kalmadan gerçekleştirebilmektedir. Bu da yaygın olarak kullanılan mobil teknolojilerle mümkündür. Çalışmadaki amacımız da öğrencilere kolay bir şekilde kullanılabilen ve her yerden erişilmesi kolay olan bir mobil uygulama ile destek olmaktır.

Eğitim alanında öğrenmenin verimli ve kalıcı bir şekilde gerçekleşmesinin sağlanması amacıyla çok sayıda uygulamalar ortaya koyulmaktadır. Bu uygulamalarda, teknik alt yapı ve öğrenme sürecinin etkili olabilmesini sağlayan içeriklerin geliştirilmesi gibi etkenler bir araya getirilerek, başarılı şekilde uygulanmasıyla hedeflenen amaca ulaşmak mümkün olabilecektir.

Bu araştırmada lisans seviyesinde öğrenim gören öğrencilerin algoritma eğitimlerine destek amaçlı IOS tabanlı cihazlara yönelik mobil öğrenme uygulaması geliştirilmiştir. Uygulama geliştirilirken yerel platform kullanılmıştır. Derleyici olarak Xcode, programlama dili olarak Swift kullanılmıştır. Veritabanındaki bilgilere erişmek için ise Core Data kullanılmıştır. Uygulama geliştirilirken, mobil uygulamayı kullanan öğrencilerin algoritma becerilerindeki değişim incelenmiştir. Mobil uygulamayı kullanan ve mobil uygulamayı kullanmayan gruba yapılan sınav sonuçlarının bulguları

incelendiğinde, gruplar arasında istatistiki olarak anlamlı bir fark olduğu görülmüştür. Mobil uygulamayı kullanan deney grubundaki öğrencilerin algoritma becerilerinin daha iyi olduğu görülmüş ve üzerlerinde olumlu etkisinin olduğu sonucuna varılmıştır. Yapılan işi kolaylaştırma, öğrencilerin başarılarındaki artış ve öğrenme kolaylığı sayesinde öğrencilerin motivasyonlarında artış gözlemlenmiştir.

Çizelge 2.6’da görüldüğü üzere çalışmaya katılan öğrencilere 6’sı test 4’ü klasik olmak üzere toplamda 10 soru sorulmuştur. Başarı sınavı; test soruları 30 puan, klasik sorular 70 puan olacak şekilde hazırlanmıştır. Deney grubundaki öğrencilerin kontrol grubundakilerden test sorularında %25 ve klasik sorularda %41.41 oranında daha başarılı olduğu tespit edilmiştir. Ayrıca, deney grubundaki öğrencilerin algoritma becerileri başarı ortalamaları 68.71 iken kontrol grubunun algoritma becerileri başarı ortalamaları 50.91 olduğu tespit edilmiş ve genelde %32.36 oranında başarı farkı olduğu anlaşılmıştır. Bu bağlamda, mobil öğrenme uygulamasını kullanan ve kullanmayan öğrencilerin sınav sonuçları arasında anlamlı bir farklılık olduğu tespit edilmiştir. Bu farklılığın ise “Algoritma Öğren” uygulamasını kullanan deney grubunun lehine olduğu görülmüştür. Sonuçlar incelendiğinde mobil öğrenme uygulaması kullanan öğrencilerin algoritma becerisi bakımından daha başarılı olduğu görülmüştür. Bu farklılığın mobil öğrenme uygulaması kullanımından kaynaklandığı sonucu çıkarılmıştır. Geliştirilen uygulamanın içerisindeki derslerin, testlerin ve alıştırmaların öğrencilerin başarılarına olumlu etki ettiği ortaya çıkmıştır.

Sonuç olarak, araştırmada geliştirilen mobil öğrenme uygulamasının lisans seviyesinde eğitim gören öğrencilerin algoritma geliştirme becerilerine katkı sağladığı sonucu çıkarılmıştır.

3.2. ÖNERİLER

Mobil teknolojiler kullanıcılarına öğrenme ve öğretme süreçleri açısından fırsatlar sunmaktadır ve mevcut öğrenci kitlesi bu yeni öğrenme türüne bilgi ve beceri bakımından hazır durumdadır. Eğitim kurumlarının ortaya çıkan yeni öğrenme yöntemlerine göre eğitim programlarını, içeriklerini ve değerlendirme araçlarını uygun bir şekilde düzenlemeleri, öğrencilerin m-öğrenme fırsatlarından daha aktif bir şekilde yararlanmalarını sağlayacaktır. Araştırma sürecinde ortaya çıkan öneriler sayesinde gelecekte bu alanda çalışma yapacak araştırmacılar için şu öneriler ortaya konulmuştur.

- Algoritma eğitimine yönelik yapılan bu çalışma, farklı işletim sistemlerini (Android, Windows Mobile vb.) kullanan mobil cihazlar için de geliştirilerek, bu işletim sistemlerini kullanan öğrenciler tarafından da kullanılabilmesi sağlanabilir.
- Uygulamanın içerisindeki testlerin ve alıştırmaların sayısı artırılabilir. Böylece öğrenci daha çeşitli sorularla karşılaşarak kendini geliştirebilir.
- Uygulama, farklı eğitim içerikleri eklenerek geliştirilebilir.
- Uygulama, kod ve tasarım bakımından yapılacak iyileştirmelerle kullanıcı deneyimini geliştirebilir.
- Uygulamaya farklı soru tipleri eklenerek test türleri artırılabilir.
- Uygulamadaki cevaplı testlerdeki sorular rastgele seçilirken, zorluk seviyesi de değerlendirilmeye katılabilir.
- Uygulamanın Şekil 2.8’de görülen veritabanındaki çıkmış soru tabloları ile kullanıcının önceden karşılaştığı sorular kaydedilerek, aynı sorularla bir daha karşılaşmaması ve soruları ezberlememesi sağlanabilir. Ayrıca bu tablolara kullanıcının soruyu çözme süreleri de kaydedilerek, sorunun ortalama çözülme süresi de belirlenebilir.
- Uygulamaya Facebook, Google vb. hesaplarla üye olabilecekleri bir sistem geliştirilerek, kullanıcıların birbirinin bilgi seviyesini görmeleri sağlanabilir.
- Uygulamaya eklenecek bir üyelik sistemiyle birbirlerine karşı yarışabilecekleri şekilde geliştirilebilir.
- Uygulamayı kullanmak için internet bağlantısı gerekmemektedir. Daha fazla içerik desteği sağlayabilmek için uygulamaya internet bağlantısı gerektiren eğitim videoları ve içeriklerin bağlantıları eklenebilir.
- Uygulama içerisine oyunlar eklenerek kullanıcıların daha eğlenceli bir şekilde öğrenmesi sağlanabilir.
- Uygulama farklı derslerde ve dillerde de kullanabilmek için geliştirilebilir.

4. KAYNAKLAR

- [1] Türkiye İstatistik Kurumu. (2016, 18 Ağustos). *Hanehalkı bilişim teknolojileri kullanım araştırması 2016* [Online]. Erişim: <http://www.tuik.gov.tr/PreHaberBultenleri.do?id=21779>.
- [2] Türkiye İstatistik Kurumu. (2017, 18 Ağustos). *Hanehalkı bilişim teknolojileri kullanım istatistikleri 2017* [Online]. Erişim: <http://www.tuik.gov.tr/PreHaberBultenleri.do?id=24862>.
- [3] Türkiye İstatistik Kurumu. (2018, 18 Ağustos). *Hanehalkı bilişim teknolojileri (bt) kullanım araştırması 2018* [Online]. Erişim: <http://www.tuik.gov.tr/PreHaberBultenleri.do?id=27819>.
- [4] Wikipedia. (2017, 10 Mart). *Mobil işletim sistemi* [Online]. Erişim: https://tr.wikipedia.org/wiki/Mobil_işletim_sistemi.
- [5] Statista. (2018, November 2). *Most popular apple app store categories in September 2018, by share of available apps* [Online]. Available: <https://www.statista.com/statistics/270291/popular-categories-in-the-app-store/>.
- [6] Statista. (2017, March 30). *Number of available apps in the apple app store from July 2008 to January 2017* [Online]. Available: <https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/>.
- [7] Statista. (2017, March 30). *Number of available applications in the google play store from December 2009 to March 2017* [Online]. Available: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>.
- [8] Anonim, Programlamaya giriş ve algoritmalar ders notları, *Ders Notları*, Kırklareli, 2007
- [9] F. Vatansever, *Algoritma Geliştirme ve Programlamaya Giriş*, 7. baskı, Ankara, Türkiye: Seçkin Yayıncılık, 2009, böl. 2, ss. 33-44.
- [10] U. E. Ayten, *Algoritma ve programlama*, *Ders Notları*, İstanbul, 2010.
- [11] J. D. Guilar and A. Loring, "Dialogue and community in online learning: lessons from Royal Roads University," *Journal of Distance Education*, vol. 22, no. 3, pp. 19–40, 2008.
- [12] D. Conrad, "E-Learning and social change: an apparent contradiction," in *Perspectives on Higher Education in The Digital Age*, M. Beaudoin ed., New York, USA: Nova Science Publishers, 2006, pp. 21–33.
- [13] M. G. Moore, "Surviving as a distance teacher," *American Journal of Distance Education*, vol. 15, no. 2, pp. 1–5, 2001.
- [14] M. G. Moore, "Three types of interaction," *American Journal of Distance Education*, vol. 3, no. 2, pp. 1–7, 1989.
- [15] D. Bouhnik and T. Marcus, "Interaction in distance-learning courses," *Journal of the American Society for Information Science and Technology*, vol. 30, no. 2, pp.

299–305, 2006.

- [16] J. L. Moore, C. Dickson-Deane, and K. Galyen, “E-learning, online learning, and distance learning environments: Are they the same?,” *Internet and Higher Education*, vol. 14, no. 2, pp. 129–135, 2011.
- [17] T. C. Reeves, K. Beaumie, K. Hyeonjin, and C. S. Loh, “Usability and instructional design heuristics for e-learning evaluation,” *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Colorado, USA, 2002, pp. 1615–1621.
- [18] P. C. Sun, R. J. Tsai, G. Finger, Y. Y. Chen, and D. Yeh, “What drives a successful e-learning? An empirical investigation of the critical factors influencing learner satisfaction,” *Computers and Education*, vol. 50, no. 4, pp. 1183–1202, 2008.
- [19] S. S. Liaw, H. M. Huang, and G. D. Chen, “An activity-theoretical approach to investigate learners’ factors toward e-learning systems,” *Computers in Human Behavior*, vol. 23, no. 4, pp. 1906–1920, 2007.
- [20] O. Serin, “Mobile learning perceptions of the prospective teachers (Turkish Republic of Northern Cyprus sampling),” *Turkish Online Journal of Educational Technology*, vol. 11, no. 3, pp. 222–233, 2012.
- [21] J. Alden, “Accommodating mobile learning in college programs,” *Journal of Asynchronous Learning Networks*, vol. 17, no. 1, pp. 109–122, 2013.
- [22] Y. C. Hsu, Y. H. Ching, and C. Snelson, “Research priorities in mobile learning: An international delphi study,” *Canadian Journal of Learning and Technology*, vol. 40, no. 2, pp. 1–22, 2014.
- [23] T. Georgiev, E. Georgieva, and A. Smrikarov, “M-learning - a new stage of e-learning,” in *International Conference on Computer Systems and Technologies*, Rousse, Bulgaria, 2004, pp. 1-5.
- [24] H. Peng, Y. Su, C. Chou, and C. Tsai, “Ubiquitous knowledge construction: Mobile learning re-defined and a conceptual framework,” *Innovations in Education and Teaching International*, vol. 46, no. 2, pp. 171–183, 2009.
- [25] G. Yokuş, “Eğitim fakültesi öğrencilerinin mobil öğrenmeye ilişkin görüşlerinin incelenmesi ve eğitim bilimleri alanına yönelik mobil uygulama geliştirme çalışması: Mobil akademi,” Yüksek lisans tezi, Eğitim Programları ve Öğretim Bölümü, Mersin Üniversitesi, Mersin, Türkiye, 2016.
- [26] S. K. Sharma and F. L. Kitchens, “Web services architecture for m-learning,” *Electronic Journal on E-Learning*, vol. 2, no. 1, pp. 203–216, 2004.
- [27] Y. Mehdipour and H. Zerehkafi, “Mobile learning for education: Benefits and challenges,” *International Journal of Computational Engineering Research*, vol. 03, no. 6, pp. 93–101, 2013.
- [28] Y. Laouris and N. Eteokleous, “We need an educationally relevant definition of mobile learning,” in *Proceedings of the 4th World Conference on Mobile Learning*, Cape Town, South Africa, 2005, pp. 290–294.
- [29] M. Bulun, B. Gülnar, and S. Güran, “Eğitimde mobil teknolojiler,” *The Turkish Online Journal of Educational Technology*, vol. 3, no. 2, pp. 165–169, 2004.
- [30] S. H. Kim, C. Mims, and K. P. Holmes, “An introduction to current trends and benefits of mobile wireless technology use in higher education,” *Association for the Advancement of Computing in Education*, vol. 14, no. 1, pp. 77–100, 2006.

- [31] M. K. Oran ve Ş. Karadeniz, "İnternet tabanlı uzaktan eğitimde mobil öğrenmenin rolü," *IX. Akademik Bilişim Konferansı*, Kütahya, Türkiye, 2007, ss. 167–170.
- [32] J. M. Spector, *Foundations of Educational Technology: Integrative Approaches and Interdisciplinary Perspectives*, 2nd ed., New York, USA: Routledge, 2015, pp. 189.
- [33] İ. Güleç, S. Çelik, and B. Demirhan, "Yaşam boyu öğrenme nedir? Kavram ve kapsamı üzerine bir değerlendirme," *Sakarya University Journal of Education*, vol. 2, no. 3, pp. 34–48, 2012.
- [34] M. G. Moore, "Toward a theory of independent learning and teaching," *Journal of Higher Education*, vol. 44, no. 9, pp. 661–679, 1973.
- [35] C. Y. Wang, B. J. Liu, K. E. Chang, J. T. Horng, and G. D. Chen, "Using mobile techniques in improving information awareness to promote learning performance," in *3rd IEEE International Conference on Advanced Learning Technologies*, Athens, Greece, 2003, pp. 106–109.
- [36] D. Parsons, "The future of mobile learning and implications for education and training," in *Increasing Access through Mobile Learning*, Alberta, Canada: Commonwealth of Learning and Athabasca University, 2014, ch. 16, pp. 217–229.
- [37] N. Özdamar Keskin, F. Z. Özata, K. Banar, and K. Royle, "Examining digital literacy competences and learning habits of open and distance learners," *Contemporary Educational Technology*, vol. 6, no. 1, pp. 74–90, 2015.
- [38] C. Kalaycı, "Dijital bölünme, dijital yoksulluk ve uluslararası ticaret," *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, c. 27, s. 3, ss. 145–162, 2013.
- [39] S. B. Shum and R. D. Crick, "Learning dispositions and transferable competencies: Pedagogy, modelling and learning analytics," in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, Vancouver, Canada, 2012, pp. 1–10.
- [40] M. Polák and L. Drápalová, "Analysis of lifespan of mobile phones : estimation of eol mobile phones generation estimation of EOL mobile phones generation," in *Electronics Goes Green 2012+*, Berlin, Germany, 2012, pp. 1–4.
- [41] J. Bosavage and J. Maselli. (2017, April 17). *Special report: Are computers destroying the earth?* [Online]. Available: <http://www.drdoobs.com/special-report-are-computers-destroying/186100362>.
- [42] D. J. Merritt, "Bias , the brain , and student evaluations of teaching," *St John 's Law Review*, vol. 82, no. 1, pp. 235–287, 2008.
- [43] D. H. Naftulin, J. E. Ware, and F. A. Donnelly, "The doctor Fox lecture: A paradigm of educational seduction," *Journal of Medical Education*, vol. 48, no. 7, pp. 630–635, 1973.
- [44] J. Danado and F. Paternò, "Puzzle: A mobile application development environment using a jigsaw metaphor," *Journal of Visual Languages and Computing*, vol. 25, no. 4, pp. 297–315, 2014.
- [45] M. Deriquito and Z. Domingo. (2017, June 28). *Mobile learning for teachers in Asia: Exploring the potential of mobile technologies to support teachers and improve practice* [Online]. Available: <http://unesdoc.unesco.org/images/0021/002162/216284E.pdf>.
- [46] P. B. Paper and I. Miyazawa, "Literacy promotion through mobile phones," in *13th*

UNESCO-APEID International Conference and World Bank-KERIS High Level Seminar on ICT in Education, Hangzhou, China, 2009, pp. 1–13.

- [47] M. Khanghah and S. H. Halili, “Design and development of mobile learning application,” *The Online Journal of Distance Education and E-Learning*, vol. 3, no. 2, pp. 31–40, 2015.
- [48] A. Kukulska-Hulme, “Learning cultures on the move: Where are we heading?,” *Educational Technology and Society*, vol. 13, no. 4, pp. 4–14, 2010.
- [49] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen, and O. Salo, “Mobile-D: An agile approach for mobile application development,” in *Companion to the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications*, Vancouver, Canada, 2004, pp. 174–175.
- [50] N. Ö. Keskin ve H. Kılınç, “Mobil öğrenme uygulamalarına yönelik geliştirme platformlarının karşılaştırılması ve örnek uygulamalar,” *Açıköğretim Uygulamaları ve Araştırmaları Dergisi*, c. 1, s. 3, ss. 68–90, 2015.
- [51] Wikipedia. (2017, March 31). *Xamarin* [Online]. Available: <https://en.wikipedia.org/wiki/Xamarin>.
- [52] Wikipedia. (2017, 31 Mart). *Qt (çalışma alanı)* [Online]. Erişim: [https://tr.wikipedia.org/wiki/Qt_\(çalışma_alanı\)](https://tr.wikipedia.org/wiki/Qt_(çalışma_alanı)).
- [53] Wikipedia. (2017, March 5). *iOS version history* [Online]. Available: https://en.wikipedia.org/wiki/IOS_version_history.
- [54] Wikipedia. (2017, 5 Mart). *iOS* [Online]. Erişim: <https://tr.wikipedia.org/wiki/İOS>.
- [55] D. Smith. (2018, June 1). *iOS version stats - April 2018* [Online]. Available: <http://david-smith.org/iosversionstats/>.
- [56] D. Smith. (2018, November 2). *iOS version stats - November 2018* [Online]. Available: <http://david-smith.org/iosversionstats/>.
- [57] S. Chacon and B. Straub, *Pro Git*, 2nd ed., Berkeley, California, USA: Apress, 2014, ch. 1, pp. 1-15.
- [58] Apple. (2017, April 17). *What is core data?* [Online]. Available: <https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/CoreData/index.html>.
- [59] Geleceği Yazanlar Ekibi. (2017, 1 Nisan). *Core data* [Online]. Erişim: <https://gelecegiyazanlar.turkcell.com.tr/konu/ios/egitim/ios-401/core-data>.
- [60] M. Gallagher. (2017, April 20). *The differences between core data and a database* [Online]. Available: <https://www.cocoawithlove.com/2010/02/differences-between-core-data-and.html>.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Fatih DİNÇ
Doğum Tarihi ve Yeri : 09.11.1986 – BOLU
Yabancı Dili : İngilizce
E-posta : fdinc14@hotmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Elektrik-Elektronik ve Bilgisayar Mühendisliği	Düzce Üniversitesi	2018
Lisans	Bilgisayar Mühendisliği	Muş Alparslan Üniversitesi	2018
Lisans	Bilgisayar Sistemleri Öğretmenliği	Süleyman Demirel Üniversitesi	2009
Lise	Bilişim Teknolojileri	İzzet Baysal Mesleki ve Teknik Anadolu Lisesi	2004